Jussi Gillberg

Targeted learning by imposing asymmetric sparsity

Department of Information and Computer Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 16.4.2010

Thesis supervisor:

Prof. Samuel Kaski

Thesis instructor:

Dr. (Tech.) Jaakko Peltonen

Aalto University
School of Science
and Technology

AALTO-YLIOPISTO DIPLOMITYÖN
SÄHKÖTEKNIIKAN KORKEAKOULU TIIVISTELMÄ

Tekijä: Jussi Gillberg

Työn nimi: Oppimisen kohdentaminen epäsymmetrisen harvuuden avulla

Päivämäärä: 16.4.2010 Kieli: Englanti Sivumäärä:8+64

Tietojenkäsittelytieteen laitos

Professuuri: Informaatiotekniikka Koodi: T-61

Valvoja: Prof. Samuel Kaski

Ohjaaja: TkT Jaakko Peltonen

Useat viime vuosina kerätyt havaintoaineistot koostuvat mittauksista hyvin pienestä määrästä näytteitä. Tällaisten aineistojen mallintaminen on haasteellista, koska mallit helposti ylisovittuvat aineistoon. Ongelmaan on kehitetty useita lähestymistapoja.

Pääasiallisen mallinnustehtävän rinnalle voidaan ottaa muita mallinnustehtäviä, joissa käytettävät mallit kytketään pääasiallisen tehtävän malliin. Näin mallien yhteisten osien oppimiseen on käytettävissä enemmän aineistoa, mikä parantaa tulosten yleistymistä uusiin aineistoihin. Tätä lähestymistapaa kutsutaan monitehtäväoppimiseksi.

Käytettävää mallia voidaan myös rajoittaa lisäämällä siihen oletuksia, jotka rajoittavat mallin sovittumista aineistoon ja siten vähentävät ylisovittumista.

Tyypilliset monitehtäväoppimista hyödyntävät mallit painottavat kaikkia oppimistehtäviä yhtä voimakkaasti, vaikka yksi oppimistehtävä on yleensä muita tärkeämpi. Tämä diplomityö on esitutkimus uudesta lähestymistavasta, joka pyrkii monitehtäväoppimisasetelmassa parantamaan yleistyvyyttä yhdessä oppimistehtävässä eri mallien sovittumiskykyä rajoittavien oletusten avulla. Valitussa oppimistehtävässä mallin sovittumista aineistoon rajoitetaan muita oppimistehtäviä enemmän mallin harvuutta lisäämällä, jotta tehtävälle opittu malli yleistyisi paremmin.

Uutta lähestymistapaa tutkitaan rajaamalla tutkimuskysymys suosittuihin LDA-malleihin, joissa hyödynnetään bayesilaisia epäparametrisia priorijakaumia. Epäsymmetrisen harvuuden vaikutuksia tutkitaan tämän malliperheen avulla. Tuloksissa on havaittavissa hienovaraisia parannuksia yleistyvyyteen. Tulokset uudella mallilla ovat kilpailukykyisiä tämän hetkisten johtavien menetelmien tulosten kanssa.

Avainsanat: bayesilaiset epäparametriset jakaumat, epäsymmetrinen monitehtäväoppiminen, harvuus, latentti Dirichlet allokaatio, pienen näytejoukon ongelmat

AALTO UNIVERSITY SCHOOL OF ELECTRICAL ENGINEERING	ABSTRACT OF THE MASTER'S THESIS

Author: Jussi Gillberg

Title: Targeted learning by imposing asymmetric sparsity

Date: 16.4.2010	Language: English	Number of pages:8+64

Department of Information and Computer Science

Professorship: Computer and Information Science	Code: T-61

Supervisor: Prof. Samuel Kaski

Instructor: Dr. (Tech.) Jaakko Peltonen

Modern data sets often suffer from the problem of having measurements from very few samples. The small sample size makes modeling such data sets very difficult, as models easily overfit to the data. Many approaches to alleviate the problem have been taken.

One such approach is multi-task learning, a subfield of statistical machine learning, in which multiple data sets are modeled simultaneously. More generally, multiple learning tasks may be learnt simultaneously to achieve better performance in each.

Another approach to the problem of having too few samples is to to prevent overfitting by constraining the model by making suitable assumptions.

Traditional multi-task methods treat all learning tasks and data sets equally, even thought we are usually mostly interested in learning one of them. This thesis is a case study about promoting predictive performance in a specific data set of interest in a multi-task setting by constraining the models for the learning tasks unevenly. The model for the data set of interest more sparse as compared to the models for the secondary data sets.

To study the new approach, the research question is limited to the very specific and popular family of so-called topic models using Bayesian nonparametric priors. A new model is presented which enables us to study the effects of asymmetric sparsity.

The effects of asymmetric sparsity are studied by using the new model on real data and toy data. Subtle beneficial effects of asymmetric sparsity are observed on toy data and the new model performs comparably to existing state-of-the-art methods on real data.

Keywords: asymmetric multi-task learning, latent Dirichlet allocation, nonparametric Bayesian statistics, small sample size, sparsity

Preface

This thesis was written during spring 2011. The work presented in this thesis was done in the Statistical Machine Learning and Bioinformatics Group of Prof. Kaski during 2010 together with the following members of the group: Ali Faisal, Gayle Leen, Jaakko Peltonen and Samuel Kaski. My work in the project was funded by the Adaptive Informatics Research Centre of the Helsinki University of Technology.

The research group is a part of Adaptive Informatics Research Centre of Helsinki University of Technology, Helsinki Institute for Information Technology, and Department of Information and Computer Science at Helsinki Institute for Information Technology.

I have been member of Prof. Kaski's group since summer 2006. Since then, I have been working on Bayesian modelling under the instruction of Dr. Jaakko Peltonen.

The research problem was formulated together. My personal contribution was to implement the new model and to design and carry out the experiments with the new model. Ali Faisal ran all the experiments with the comparison method.

I would like to thank Prof. Kaski and Dr. Peltonen for the professional guidance that I have recieved during the past years, and especially for the interesting conversations about science. A special thanks goes to Dr. Gayle Leen, whose help in the most difficult phase of the project was indispensable and without whom this research project would never have produced any results.

Otaniemi, 15.5.2011

Jussi Gillberg

Contents

Symbols and abbreviations

Symbols

$\boldsymbol{a}$	Vectors are denoted using bold case.
$\boldsymbol{A}$	Matrices are denoted using bold uppercase.

Operators

$a \cdot a$	Product of the elements a and b
$\sum_i$	Sum over the index i
$\prod_i$	Product over the index i
$\mathbf{a}.\mathbf{b}$	Hadamard product of the vectors $\mathbf{a}$ and $\mathbf{b}$

Abbreviations

IBP	Indian buffet process
IBPCD	Indian buffet process compound Dirichlet process
HDP	Hierarchical Dirichlet process

1 Introduction

Statistical machine learning is a field of computer science in which statistical models are fitted to data sets. The purpose of modeling is to learn about the phenomena described by the data sets and to be able to make predictions about future observations. Methods of statistical machine learning are used in other fields to discover new knowledge. For example in bioinformatics, data sets with measurements from living cells are modeled to explore the role of genes under different medical conditions and treatments. Modeling has proved its value as means of discovering knowledge, but modeling methods lag behind the challenges related to many modern data sets [1].

One of the central challenges of modern data sets in bioinformatics is that often they consist of measurements from very few replicates, as producing replicates tends to be very expensive. Simultaneously the number of different attributes measured from the few replicates has exploded. For example in the case of gene expression measurements, the expression of 10000-40000 genes is usually measured simultaneously from each replicate, whereas the number of replicates is atmost 100-200. The setup of having more measured attributes than replicates is known as the *small n, large p* problem, where n denotes the number of replicates and p the number of measured attributes or dimensionality of the data.

Having more measured attributes than replicates renders most methods of traditional statistics useless due to statistical issues. In statistical machine learning, models are often constrained with further assumptions that allow tackling problems such as the *small n, large p* problem. One such assumption is that of *sparsity*, in which observed patterns in the data are assumed to be related to changes in the activity of relatively few attributes or factors (see e.g. [2]). For example in the case of bioinformatics, we might assume that a disease is characterized by a dramatic change in the expression of only a few genes instead of a huge set of small changes in thousands of genes.

Another solution to the problem of having too few samples in the data set of interest is to include other, secondary sources of data, which do not match the primary data set of interest perfectly but which are still representative. Secondary data can be made use of in multiple ways. In *multi-task learning* (see e.g. [3]), secondary data sets are modeled with data-set-specific models, and some parts of these models are shared with the model for the primary data set. Given that the data sets share characteristics that can be learnt using shared resources, more data is available for learning the shared parts. Reducing uncertainty about the shared parts allows also better modeling of the data-set-specific parts and overall better predictions.

In most applications we are often interested in making predictions about a particular distribution described by a *data set of interest*. In other words, our interest is asymmetric in the sense that we are not equally interested in all data sets. We take the multi-task learning approach in order to augment the data set of interest with secondary data. Even though we are more interested in some parts of our data than the others, traditional multi-task learning treats all data sets and models symmetrically. The models can be constrained by making further assumptions to concentrate their resources on the data set of interest. This is called *asymmetric multi-task learning* [4]. This thesis is a case study on using sparsity assumptions for implementing asymmetric multi-task learning setups.

The aim of this thesis is to study whether predictive performance of the model for a data set of interest could be increased in a multi-task setting by making it more sparse compared to models for secondary data sets. The question is naturally too broad and complex to be addressed completely within the scope of this thesis, so the research question is limited to the very specific and popular family referred to as topic models, using Bayesian nonparametric priors.

The general Bayesian framework, under which the proposed model has been formulated, will be presented in section 3. Bayesian nonparametrics and topic models will be presented in sections 4 and 7 respectively. To empirically study the research question, an implementation of the model presented in section 8 is used to conduct experiments, which are described in section 9. Section 10 contains discussion about the research problem and the results.

2 Modeling and data analysis under the *small n, large p* -problem

Modeling is about finding structure in data by fitting models into data sets. Models can be used to to make predictions about future observations or to summarize data sets. Modeling is a very broad term and the variety of different models is extensive. Examples of modeling techniques include regression models, clustering algorithms for grouping similar observations together and simulation models for physical processes. This thesis concentrates on the field of statistical modeling, in which data sets are described using probability distributions.

As mentioned in Section 1, modern data sets collected, for example, in the fields of bioinformatics and neuroinformatics often suffer from the problem of having too few data. The number of measured attributes per sample (biological replicate or patient) is huge, whereas the number of samples tends to be very small. This problem is referred to as the *small n, large p* -problem, where n denotes the number of samples and p the number of measured attributes. The small n, large p problem poses great challenges for modeling.

The common inbalance between the number of samples and measured attributes is due to development in measurement technologies that allow measuring thousands of attributes simultaneously from each sample. As the economical cost of each individual sample is high, data sets consist of thousands of measurements from very few samples.

The severity of the small n, large p problem varies in different fields. For example gene expression is usually recorded for $10000 - 40000$ genes whereas the number of samples in such experiments is $10 - 100$. In neuroinformatics the number of samples is of the same order, but the number of measured attributes (voxel activities) is of the order of 1000000. However, the severity of the problem is also affected by other properties of the learning task in addition to the ratio of measured attributes and samples. As an example, in neuroinformatics the spatial relationship between different voxels is known allowing effective treatment of the problem, whereas in bioinformatics the relationships between the genes are often poorly understood. Even though the severity of the small n, large p problem varies, it is most definitely the main challenge for most methods used in the affected fields.

The small n, large p problem renders many traditional methods useless. For example, the estimation of the parameters for a simple regression model using classical techniques becomes impossible for statistical reasons as the number of variables exceeds that of samples. The severity of the effects of the small n, large problem depend on the state of ill-conditionedness: before rendering estimation of parameters impossible, parameter estimates become more and more unreliable as the ratio of samples and measured attributes gets worse.

The small n, large p problem is strongly related to the problem of *overfitting*. When using a model to make predictions, the model parameters are fitted to some *training data*. Overfitting refers to models generalizing poorly to new data sets that have been generated from the same distribution as the training data. The small n, large p problem promotes overfitting.

2.1 *Small n, large p* -problem and overfitting

When modeling, data are assumed to have some structure that the model can capture. Learning the structure will enable making predictions about future observations. If a model is flexible enough, it can, however, learn the noise in the training data in addition to the structure with predictive power. Learning the noise pattern will cause a reduction in predictive performance, as noise does not contain any predictive power. Overfitting occurs in both supervised and unsupervised learning.

For example in regression analysis, some recorded noise in the finite data set can be correlated with the target variable more strongly than such attributes that in reality have predictive power. In these cases the model will learn to make predictions based on the noise, and the model will disregard the interesting structure in the data. Ill-posed data sets suffering from the small n, large p problem are especially prone to overfitting, as the large number of attributes available for making predictions induces a huge number of model parameters in most traditional models, and the huge number of model parameters enables fitting extensively to the finite training data.

The small n, large p problem and the problem of overfitting are easiest to describe by using a simple example. Assume that we want to learn to predict, whether a child of five years becomes a Doctor of Technology later in his/her life. Our data set consists of features about a group of 100 people with information about them at the age of 5 and their academic status at the time of their death.

First let us assume that the data set contains the following information for each of the 100 subjects: parents' academic status, family income level and the subjects' preference of chocolate ice cream vs. strawberry ice cream at the age of 5 years. Obviously parents' academic status predicts the academic status of the subjects, but not to perfectly. Also family's income level will have some predictive power, even though not as much as parents' academic status. Ice cream preferences, however, will probably not contain any information about future academic status. With 100 subjects and 3 attributes, we could use a standard logistic regression model to learn to make predictions about academic status. The learnt model would probably not explain the training data perfectly, but it would probably generalize well to new observations.

Now lets assume that in addition to the previous information, we have also recorded the favorite color at the age of 5 and the first letter of the middle name for all subjects. These new attributes are unlikely to contain any predictive power about future academic status, but the parameter estimates will have more noise than without these new, noninformative variables as the addition of excess, noninformative parameters causes overlearning. Performance in the training set will increase, but the model will not generalize as well as without the excess variables to new observations.

By adding more and more noninformative features the parameter estimates will become more and more unreliable. Eventually some set values of the noninformative features will correlate perfectly with the future academic status. For example, all the 100 people who will become Doctors of Technology might favor chocolate ice cream, have a middle name starting with 'T' and have an uncle with eye glasses. This set of characteristics will probably tell nothing about future observations, but in this finite set of 100 samples it is able to distinguish the set of people who will become Doctors of Science. At this stage learning the parameters will go wrong: the model will learn this set of features allowing perfect performance in the training set and probably ignore parents' academic status, as it is no longer needed with its partial predictive power. This set of parameters chosen by the learning algorithm suffers from severe overlearning, as the model has learnt the noise and missed the true structure with predictive power completely.

The problem can, however, get even worse. If the number of noninformative features is increased even further, the model will soon have many different ways of distinguishing the future Doctors of Science. Selecting some unique parameters without further information becomes impossible. This corresponds to the worst case scenario of the small n, large p problem, in which learning parameters for a model without, for example, further regularization becomes impossible. Regurlarization is discussed in sections 2.2 and 6.

More generally, as the number of features increases, the number of feature combinations allowing good performance in the training set increases. Any single solution will probably overfit and generalize poorly to the test data. This can be alleviated by taking the Bayesian approach as will be discussed in section 3.2.

2.2 Previous work on the *small n, large p* -problem

In statistics, the *problem of multiple testing* has been studied extensively [5]. The problem of multiple testing deals with testing multiple statistical hypotheses simultaneously and it is closely related to the small n, large p problem.

In traditional statistics, hypotheses about the structure of the data are made. These hypotheses are then tested and potentially accepted based on *test statistics*. Hypotheses are, for example, of the form: "is the average height of men greater than that of women?" The *null hypothesis* refers to a hypothesis that will be accepted when other hypotheses are rejected. In the case of the previous example, the null hypothesis is "the average height of men and women is the same."

Test statistics are computed from the data and they can be used to assess how much evidence there is for different hypotheses. An example of a test statistic is the difference of average heights of the men and women in the test sample. Based on the assumptions about the data, test statistics are random variables with corresponding distributions.

Evidence for different hypotheses is evaluated by assessing how probable the value of the test statistic is based on the different hypotheses. As test statistics are random variables, there is always a risk of accepting a hypothesis even though some other hypothesis holds true. For example, we might accept a hypothesis about the differing means of two sets of observations even though the observation sets are generated from the same distribution.

The threshold of the test statistic for accepting a hypothesis other than the null hypothesis is usually set by first selecting a *risk level*, which is the probability of accepting a hypothesis when the null hypothesis is true. The selected risk level then defines the value of the test statistic which is used as a threshold for hypothesis acceptance/rejection. For example, we might choose to accept the hypothesis of the average heights of men and women differing when the test statistic has a value, which is produced with a probability of α when the null hypothesis actually holds true. A *p-value* is the probability of the value of the test statistic being at least as extreme as the one computed from the data, assuming that the null hypothesis is true.

The error of accepting a hypothesis even though the null hypothesis is true is referred to as a *false positive*.

Hypotheses can be made about the roles of single variables (*univariate methods*) or groups of variables (*multivariate methods*). In both cases, as the number of measured attributes increases, the number of hypotheses will increase and also the number of false positives will increase.

To reduce the number of false positives, we can require more evidence before accepting hypotheses. If hypotheses are accepted based on their P-values, the requirement for extra evidence can be implemented simply by making the threshold of the P-values for accepting hypotheses higher. By making some assumptions, thresholds for accepting hypotheses can be computed in a rigorous way. An example of this is computing the *Bonferroni correction*. If a false positive rate of α is desired, it can be shown that by using a p-value threshold of α/n, where n is the number of hypotheses to be tested, the desired overal false positive rate can be attained. The assumptions made for the Bonferroni correction do not, however, in practice usually hold, and due to this the method often gives too strict acceptance thresholds [6].

One of the most studied approaches to alleviating the *small n, large p* -problem is *regularization* of model parameters. Model parameters are learnt by optimizing some objective function. Regularization refers to constraining the parameters used to make predictions or to penalizing the objective function for extreme values of parameters.

Regularization helps by limiting the set of potential solutions in the optimization problem of fitting model parameters. Regularization can be seen as imposing Occam's razor on the solution: each regularization method includes some measure for the complexity of the solution, which is then used to restrict the set of possible parameter combinations to be used for predictions. For example in the extreme case of the small n, large p problem, where a unique solution can not be found as infinitely many sets of parameters can explain the training data, regularization can constrain the set of possible parameters enough to allow finding one unique solution.

Parameters can be regularized in multiple ways. $L2$ regularization penalizes heavily solutions, in which some parameters have large absolute values. $L1$ regularization suppresses most parameter values to 0, promoting the model to use only a subset of possible parameters for making predictions. This enforces *sparsity* in the parameter space[2]. Differences between $L1$ and $L2$ regularization are discussed more in section 6.

The idea of sparsity can be extended further. Assumptions that suppress weak effects and instead use a few strong effects to explain the data are commonly used in machine learning. Section 6 explores sparsity more generally.

Another well-established approach for alleviating the small n, large p problem is *multi-task learning*. In multi-task learning other sources of data are used to provide more evidence. Additional data is not assumed to be generated by the same distribution that has generated the data set of the interest, and therefore models must take into account this deviation. Section 5 describes the multi-task learning approach in more detail.

3 Bayesian framework

The contribution of this thesis has been formulated in the general framework of Bayesian statistics. In Bayesian statistics it is assumed that model parameters are random variables with corresponding probability distributions.

In Bayesian statistics the probability of an outcome of an event can be considered as a measure of belief for different outcomes as opposed to *frequentist* statistics, where probability is more strictly considered as the ratio of outcomes of a random process. Under the Bayesian framework probabilities can, for example, be computed for deterministic outcomes when information about the outcome is insufficient to deduce the true deterministic result.

The Bayesian framework is one of the frameworks that enables probabilistic modeling. Probabilistic modeling allows quantifying the amount of uncertainty related to learnt models. This is crucial in decision making.

The major drawback of Bayesian statistics is that models tend to be computationally expensive. Due to this, Bayesian methods have only become feasible for large scale analysis during the last 30 years.

In sections 3.1 - 3.3 the essential concepts and techniques of the Bayesian framework are presented. Section 3.4 introduces the idea of hierarchical modeling which is largely responsible for the success of Bayesian modeling. Section 3.5 presents the elementary probability distributions needed for understanding the novel model presented in Section 8.

3.1 Bayes' theorem

The process of Bayesian modeling starts with the selection of the *model family* M which defines the functional form of $p(data|\Theta, M)$. Usually the model family is omitted from the equations, when we are only dealing with one model family, $p(data|\Theta, M) = p(data|\Theta)$. A model family has a set of parameters, together denoted with Θ, whose values define the actual model. Examples of model families are the Gaussian distribution, $data \sim \mathcal{N}(\Theta)$ or a mixture of Gaussians, $data \sim \sum_i \pi_i \cdot \mathcal{N}(\Theta_i)$, with parameter sets $\Theta = \{\theta\}$ and $\Theta = \{\theta_i, \pi_i\}_i$ respectively.

Traditional modeling often aims to maximize the *data likelihood*, that is to look for such parameters Θ in the model family that describe the training data as well as possible given the selected model family. In Bayesian statistics, the information in the data is augmented by additional information about the distribution of parameter values. Knowledge concerning the parameters is incorporated into a *prior distribution*. For example, the model parameters Θ of a simple Gaussian model could be known to be distributed according to a Gaussian distribution with parameters $0, 1$, in other words $\theta \sim \mathcal{N}(0, 1)$.

After defining the model family of the data likelihood $p(data|\Theta)$ and the prior distribution $p(\Theta)$ for the model parameters, the joint probability distribution of the parameters and the data can be computed:

$$p(data, \Theta) = p(data|\Theta) \cdot p(\Theta). \tag{1}$$

In the observed data we get more information about the distribution of the parameters. This information is incorporated in the joint distribution when the prior is multiplied with the likelihood.

As we are interested in the conditional distribution of the model parameters instead of the joint of model parameters and data, the joint distribution is transformed into a conditional distribution by dividing it with the *marginal distribution* of the data,

$$p(Data) = \int_{\Theta} p(data|\Theta) \cdot p(\Theta) \; d\Theta. \tag{2}$$

The conditional distribution of the model parameters that results from combining the new information from data to the prior knowledge is referred to as the *posterior distribution*, and it summarizes current the information about the distribution of model parameters contained in the prior distribution and the data likelihood. The posterior can be computed by applying the Bayes formula

$$p(B|A) = \frac{p(A|B) \cdot p(B)}{p(A)} \tag{3}$$

where $p(A) = \int p(A|B) \cdot p(B) \; dB$. This yields the posterior distribution

$$p(\Theta|data) = \frac{p(data|\Theta) \cdot p(\Theta)}{p(data)}. \tag{4}$$

The main challenge in Bayesian modeling is the computation of equation 4 [7], which is discussed in more detail in section 3.3. Data can be summarized using quantiles and other statistics of the posterior distribution. The interest is, however, more often in making predictions by using the *posterior predictive distribution*.

3.2 Posterior predictive distribution

The most important use of the posterior distribution is through the *posterior predictive distribution*

$$p(data_{new}|data) = \int_{\Theta} p(data_{new}|\Theta) \cdot p(\Theta|data) \; d\Theta, \tag{5}$$

which can be used to make predictions about future observations. The posterior predictive distribution allows taking into account our current uncertainty about the model parameters when making predictions about future observations $data_{new}$.

Making predictions based on the whole posterior distribution instead of using any single model parameters is advantageous when there is uncertainty (see e.g.[8]). When using the posterior predictive distribution, predictions are made by integrating over the posterior and weighting different parameters according to their posterior proabability.

3.3 Posterior computation

The posterior distribution quantifies the strength of belief over different values of model parameters. It combines prior information and information in data to summarize current information.

Most often solving the posterior analytically is intractable [7]. Especially solving the integral required to compute the marginal distribution of the data in equation 2 tends to exceed the integration skills of even the most experienced researchers if complicated enough models are used.

It is important to note that being able to compute single values of the posterior density function is not enough. Single values of the posterior will not tell, how well the current set of values computed from the posterior describes the entire posterior distribution. A set of values can correspond to very unlikely parameter values, and the main posterior mass can be completely undiscovered. This problem is ubiquitous with models with a huge parameter space: studying the parameter space for example by computing a grid of posterior values quickly becomes computationally impossible as the parameter space grows exponentially with respect to the number of model parameters.

Different numerical approaches exist that allow computing an approximation for the posterior. Numerical methods generate *samples* from the distribution of interest. The distribution of the samples corresponds to the distribution of interest. Samples can then be used to compute quantities of interest.

3.3.1 MAP estimates

The simplest numerical approach is computing the *maximum a posteriori* (MAP) estimate, which is the parameter value that maximizes the posterior probability.

One of the problems with MAP estimates is that a single value does not describe the level of uncertainty related to a posterior distribution. Using a single value instead of the full posterior distribution can produce very different predictions. This holds true especially when the posterior distribution is spiky. For example, a parameter might have a posterior distribution with a very narrow region with high posterior density, while most of the posterior mass lies outside this area. Making predictions using a MAP estimate would in this case produce predictions that reflect poorly the complete posterior, which contains all current information about the distribution [7].

3.3.2 MCMC methods

Markov Chain Monte Carlo(MCMC) methods have been used extensively during the last decades for posterior computation.

The idea of MCMC methods is to construct a first order Markov chain, whose *stationary distribution* is the distribution of interest $p^*(z)$. A first order Markov chain is a random variable whose state distribution at time t depends only on the previous state z_{t-1} of the chain. A stationary distribution is a state distribution that is not changed by random transitions defined by the Markov chain.

The state of the Markov chain is updated by generating random transitions using transition probabilities $T(z_i|z_{i-1})$. A *sufficient* condition for the chain to converge to the distribution of interest is the *detailed balance condition* defined by

$$p^*(z) \cdot T(z, z') = p^*(z') \cdot T(z', z).$$

This is referred to as the chain being *reversible*. The detailed balance condition requires that the chain is *unique*, which is quaranteed when the transition probabilities remain unchanged at all times t.

After the construction of a Markov chain with the desired stationary distribution, the state of the chain is updated iteratively. When all the assumptions mentioned above hold, the chain is known to converge to the distribution of interest [2]. To be more precise, the distribution of the states z_t of the chain is known to converge to the distribution of interest.

MCMC methods are often referred to as *random walk* algorithms. The algorithms proceed by taking random steps in the parameter space according to some probability distribution computed by using the samples from the previous step(s).

Two MCMC methods are used to compute posterior distributions for the novel method developed in this thesis. The two methods are called *the Metropolis-Hastings algorithm* and *Gibbs sampling*.

Gibbs sampling and the Metropolis-Hastings algorithm are based on being able to compute the posterior distributions of parameters (or groups of parameters) given the values of other parameters. In other words, the conditional posterior distributions of parameters are of a form from which it is possible to generate samples easily. The algorithms operate by generating samples from the conditional posterior distributions of different parameters (or groups of parameters) one at a time and then using the new values of the parameters to generate posterior samples from the distributions of other parameters. The distribution of the generated sample will then converge to the posterior distribution, if sampling is continued indefinitely. The Metropolis-Hastings algorithm and Gibbs sampling are described in more detail in sections 3.3.3 and 3.3.4.

Construction of Markov chains used by MCMC methods is feasible when it is too difficult or inefficient to generate samples from the posterior distribution directly. The problem is that even though convergence is guaranteed when sampling infinitely, it is difficult to assess whether the chain has converged or not.

Methods for assessing convergence exist[7]. These methods can be used to indicate that the chain has not converged but they can not guarantee convergence.

In practice, the question of convergence is often paid little attention to. Theory of MCMC methods states that as the number of samples generated from the posterior increases, their distribution will better approximate the distribution of interest. The number of posterior samples generated is usually, however, chosen according to computational resources, and the approximation they yield is taken as the best possible approximation available. This is approach is also taken in this thesis.

Consecutive samples produced by the MCMC methods are correlated. Due to this, consecutive samples do not contain much information about the posterior as compared to a single sample. To save memory and reduce the computational expense, MCMC chains are often *thinned.* Thinning means that every k^{th}

3.3.3 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm involves two components: (1) *a proposal distribution* used to propose steps for the random walk in the parameter space and (2) an acceptance/rejection rule used to select whether the proposals are accepted as samples or not.

The Metroplis-Hastings algorithm proceeds as follows:

1. The algorithm is initialized by randomizing or selecting some set of initial values Θ^0 for model parameters.

2. Sample a *proposal value* Θ^* from the proposal distribution $J(\Theta^*|\Theta^{t-1})$ based on the value of the previous accepted sample Θ^{t-1}.

3. Compute
$$r = \frac{p(\Theta^*|data, prior)/J(\Theta^*|\Theta^{t-1})}{p(\Theta^{t-1}|data, prior)/J(\Theta^{t-1}|\Theta^*)}$$

4. Set
$$\Theta^t = \begin{cases} \Theta^* & \text{with probability } \min(r, 1) \\ \Theta^{t-1} & \text{otherwise} \end{cases}$$

Steps 2-4 are iterated until the desired number of samples has been obtained.

Often the bottleneck in the computation of the posterior distribution $p(\Theta^*|data, prior)$ is the normalization term in the denominator of the Bayes formula (equation (4)). Computing the integral involved tends to be infeasible. The Metropolis-Hastings algorithm solves the problem by comparing two samples from the posterior distribution so that the normalization constant (denominator) cancels out.

If the proposal distribution in the Metropolis-Hastings is ill-suited, the acceptance rate can become too low. Then the chain will not traverse the parameter space and the estimate for the posterior distribution will be poor [7].

3.3.4 Gibbs sampling

Often it is possible to compute the exact conditional posterior distribution of a parameter given the values of all other parameters. In these cases the parameters can be updated in an iterative fashion without rejecting any part of the samples. This is referred to as Gibbs sampling.

Gibbs sampling for a model with two parameters with conditional posteriors of a known form proceeds as follows

1. Initialize some set of initial values $\{\theta_1^0, \theta_2^0\}$ for model parameters by randomization or selection (e.g. use the MAP estimate).

2. Sample parameter θ_1^t from the distribution $p(\theta_1^t|data, \theta_2^{t-1})$.

3. Sample parameter θ_2^t from the distribution $p(\theta_2^t|data, \theta_1^t)$.

Iterate steps 2 and 3 until the desired number of samples has been generated.

Mathematically Gibbs sampling corresponds to a Metropolis-Hastings algorithm which uses the true conditional posterior as a proposal distribution resulting in an acceptance rate of 1 [7].

3.3.5 Rejection sampling

Rejection sampling allows generating samples from a distribution $p(\theta)$ by using a (possibly unnormalized) proposal density $g(\theta)$. Rejection sampling requires that a positive proposal density $g(\theta)$ is known for all θ for which $g(\theta) > 0$ when $p(\theta) > 0$ and that $g(\theta)$ has a finite integral. Also the *importance ratio* $p(\theta)/g(\theta)$ must have a known bound M: $p(\theta)/g(\theta) \leq M \ \forall \ \theta$.

The algorithm proceeds as follows:

1. Generate a sample from $g(\theta)$

2. Accept θ with probability $p(\theta)/(M \cdot g(\theta))$, otherwise return to 1.

Accepted values will follow the correct distribution [7].

3.4 Hierarchical models

The success of Bayesian modeling is strongly based on the possibilities of *hierarchical modeling*. In hierarchical modeling parameters are bound together by connecting them through a shared prior.

Hierarchical modeling allows *sharing of statistical strength* between parameters. When little data is available for learning a set of parameters but the parameter values are known to be similar, by connecting them with a shared prior the parameters will share the information available for learning each with the other parameters through the shared prior.

An example about a case in which hierarchical modeling could be useful is the following: assume that we want to learn the average height of men in Finland and Germany, but we only have 20 measurements from each of the countries. A set of 20 samples is by far too small and will easily give unrealistic results: a few exceptional measurements about very short or long people will deviate the mean from the true population average. The problem can, however, be alleviated by using a hierarchical model.

A hierarchical model for the problem would assume that the parameters used for modeling the average height in the different countries have a shared prior distribution describing the average height of, for example, European men in general. The mean parameter for this distribution would be learnt based on the region-specific means. By varying the variance of the prior for region-specific means, the region-specific means could be forced to resemble each other more or less. This corresponds to assuming that the region-specific means are similar, though not identical. In practice, binding the region-specific means together would help reduce the effects of outliers in the data sets and give more realistic results [7].

Hierarchical modeling is ubiquitous in multi-task learning, which is presented in section 5. The new model developed in section 8 is a Bayesian hierarchical multi-task model.

3.5 Elementary probability distributions

The elementary probability distributions used in the novel method developed in this thesis are presented in this section.

Selection of prior distributions is heavily affected by the requirement of the property of *conjugacy*. Most elementary distributions have a *conjugate prior*. A conjugate prior of probability distribution A is such a distribution, that multiplication of likelihood terms from distribution A will produce a posterior that has the same form as the as prior distribution. In other words, the posterior distribution will have the same form as the prior as long as the prior is the conjugate prior of the likelihood function [7].

Using conjugate priors often makes computations easier. The use of conjugate priors allows posterior distributions to have a well-known form, from which it is possible to generate samples directly. This enables Gibbs sampling and other effective ways of computation.

3.5.1 Multinomial and Dirichlet distributions

The Dirichlet and multinomial distributions are used in topic models to model counts of words. The multinomial and Dirichlet distributions are important for this thesis as the model developed uses the topic models framework. Topic models will be described in more detail in section 7.

The multinomial distribution can be used to assign a probability for the outcome of n trials where each of the trials results in exactly one of the k (fixed and finite) outcomes. By denoting the random variable for the number of successes in each of the k outcomes with X_k and the observed number of successes by x_k this probability can be written as $p(X_1 = x_1, \dots, X_k = x_k)$.

The probability density function of the multinomial distribution is

$$f(x_1, \dots, x_k; n, p_1, \dots, p_k) = \left(\begin{array}{c} n \\ x_1 \cdots x_k \end{array} \right) p_1^{x_1} \cdots p_k^{x_k}, \tag{6}$$

where $\sum_{i=1}^{k} x_i = n$. The parameters of the multinomial distribution $(p_1, \dots, p_k)$ correspond to outcome probabilities, and naturally $p_i \in [0, 1]$ and $\sum_{i=1}^{k} p_i = 1$.

The multinomial distribution only models the counts, not the order of their occurrence. This is crucial in topic models, where documents are treated as bags of words and word orderings are disregarded.

The conjugate prior of the multinomial distribution is the Dirichlet distribution, which is a probability distribution that generates probability measures. In other words, the Dirichlet distribution can be used to compute the probability of observing a set of k elements p_i, where $p_i \in [0,1]$ and $\sum_{i=1}^{k} p_i = 1$, that can be interpreted as the probabilities of k (fixed and finite) exclusive events. Therefore the Dirichlet distribution can be used to generate the parameters used by the multinomial distribution.

The probability density function of the Dirichlet distribution is

$$f(x_1,\ldots,x_K;\alpha_1,\ldots,\alpha_K) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \tag{7}$$

where the parameters $\alpha_1,\ldots,\alpha_K$ correspond to prior observations about outcomes and $\mathrm{B}(\alpha)$ is the beta function. The beta function can be written as

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\bigl(\sum_{i=1}^{K} \alpha_i\bigr)}$$

using Γ to denote the gamma function. [7]

3.5.2 Bernoulli and beta distributions

Bernoulli and beta distributions are important components of the Indian Buffet Process prior (abbreviated IBP) that is presented in section 4.2.3. The IBP is used in the new method developed in this thesis. The Bernoulli distribution is also used in the new model to impose asymmetric sparsity.

The Bernoulli distribution is the distribution of the result of a single trial with two possible outcomes: success and failure.

The probability density function of the Bernoulli distribution is

$$f(k;p) = p^k (1-p)^{1-k} \quad \text{for } k \in \{0,1\}, \tag{8}$$

where p denotes the probability of success. The probability of success is obtained by setting $k = 1$ and the probability of failure by $k = 0$.

The conjugate prior of the Bernoulli distribution is the beta distribution. The beta distribution can be used to compute a likelihood for a probability given parameters α and β that denote the prior observations of the outcomes of the process [7].

The probability density function of the beta distribution is

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}. \tag{9}$$

3.5.3 Negative binomial distribution

The negative binomial distribution is used for modeling the number of words in a document in the new model developed and in *focused topic models* that is the most similar existing approach. This use will be presented in sections 8.3 and 8.5.2.

The negative binomial distribution is a distribution for the number of successes in a sequence of Bernoulli trials with parameter p before observing r failures.

The probability density function of the negative binomial distribution is

$$f(k) = \binom{k+r-1}{k}(1-p)^r p^k \quad \text{for } k = 0, 1, 2, \ldots. \tag{10}$$

[7]

3.5.4 Gamma distribution

The gamma distribution is used in the *IBP compound Dirichlet process* (abbreviated IBPCD) prior, which is a part of the novel model presented in this thesis. The IBP compound Dirichlet process is presented in section 4.2.4.

Gamma distribution is a two parameter distribution (shape k, scale θ). When the shape parameter k is an integer, it corresponds to the distribution of the sum of k exponentially distributed random variables with mean parameters θ. When k is not an integer, the distribution does not have any clear interpretation [7].

The probability density function of the gamma distribution is

$$f(x; k, \theta) = x^{k-1}\frac{e^{-x/\theta}}{\theta^k\,\Gamma(k)} \text{ for } x \geq 0 \text{ and } k, \theta > 0, \tag{11}$$

where Γ denotes the gamma function.

3.5.5 Poisson distribution

The Poisson distribution is required for understanding the IBP and IBPCD priors. The IBPCD is used in the novel method developed, and the IBPCD builds on the IBP. The IBP and IBPCD are presented in sections 4.2.3 and 4.2.4 respectively.

The Poisson distribution is the distribution for the number of events k occurring in a fixed period of time if these events occur with a known average rate λ and independently of the time since the last event [7].

The probability density function of the Poisson distribution is

$$f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{12}$$

where e denotes the base of the natural logarithm.

4 Model selection and nonparametric Bayesian statistics

Selection of the model family used for modeling data is one of the foremost challenges of statistical modeling. This is usually referred to as *model selection*. Examples of model families are the Poisson distribution, a Gaussian distribution and a mixture of 2 Gaussian distributions. The model family defines which data sets the model will be able to describe successfully. When starting to model a new real data set, it is not known in advance which models will perform well.

The more powerful (complex) a model is, the better it is able to fit to any given training data. Therefore performance on training data can be increased by adding model complexity.

If the model used is too complex, overfitting can occur. Given too much resources, a model will learn the recorded random noise in the training data in addition to structure having predictive power. In other words, the model will learn to use recorded noise values for predicting quantities of interest. This will harm performance, as noise in training set does not have any predictive power on future observations and therefore making predictions based on it will cause error. Overfitting is discussed in more detail in section 2.

As performance on training data will always increase by increasing model complexity, performance of models needs to be studied using some other data than that used in training model parameters. Using these estimates for performance model selection can be done.

4.1 Cross validation

One of the most common approaches to model selection and model performance evaluation is *cross validation* [2]. Cross validation is used to assess performance of the new model developed in this thesis.

In cross validation, the data set of interest is divided into distinct subsets. Model parameters are fitted using all but one of the subsets as the *training set* and model performance with the learnt parameters is validated on the remaining subset, which is referred to as the *validation set*. This process is repeated until all subsets have been used as a validation set. To get an estimate on model performance, an average over the results on the validation sets is taken.

The number of subsets varies according to circumstances: computational expense related to learning the model parameters and the amount of computational resources available define a sensible division.

Division is done randomly in order to break potential structure that could affect cross validation results. For example, the person collecting the data set could have included measurements into the data set in the order of generation. Conditions such as fatigue could have resulted in the quality of data decaying towards the end of the data collection procedure. Random assignment of samples into the different subsets will help avoid such sources of bias.

The estimate of the model performance can be used in model selection. The model family (complexity) that produces the best predictions on validation data is chosen. Parameters will possibly overfit to the training set but the selection is done based on the performance on validation set, to which the parameters have not been able to fit at all. Models that have overfitted severly will have worse performance on the validation data than models that have not learnt to make predictions based on recorded noise.

It is, however, possible to overfit to the validation sets in the specific division of data. Therefore in order to get a better estimate of the model performance on new data, it is sensible to divide the data set in the beginning of the model selection process to the training and validation set and to the *test set*. After model selection by cross validation, model parameters are learnt again using the combined data from training and validation sets. Performance of the model is then measured in the test set. The aim of this is that none of the model parameters (including model complexity) is fitted to the actual test set. Therefore performance on test set corresponds to performance on new data, to which model parameters have not been fitted.

4.2 Nonparametric Bayesian methods

Nonparametric Bayesian methods present an alternative approach to model selection. They assume such a prior distribution for model complexity that model complexity can be treated as a parameter among others. Often the prior distribution for complexity emerges as a result of other assumptions and it is not written out explicitely. In other words, these methods often assume an implicit prior distribution for model complexity.

The motivation for using nonparametric Bayesian methods is that model complexity becomes a parameter among others which can be sampled instead of, for example, computationally expensive testing in a validation set.

4.2.1 Dirichlet process

The Dirichlet process (DP) is the best-known nonparametric Bayesian prior. First publications about using the DP as a prior to avoid model selection are [9] and [10] and they date 40 years back. The DP was later extended to the Hierarchical Dirichlet Process that is used as a comparison method for the new model presented in this thesis.

The DP is a prior for the exhibition of a possibly countably infinite number of atoms. In addition to the atoms, the DP generates a sequence of exponentially decreasing probabilities corresponding to the atoms.

The atoms and their associative probabilities can be used as a prior for model structure. For example, the atoms can be used as cluster centers and the probabilities as a prior for cluster memberships. More specific technical and mathematical details of the Dirichlet process are omitted here, as understanding them is not crucial for understanding this thesis.

4.2.2 Hierarchical Dirichlet Process

One of the best known nonparametric models was presented in [11] in 2007. The *Hierarchical Dirichlet Process* (HDP) extends the Dirichlet process to a hierarchical case, where multiple Dirichlet processes share a common Dirichlet process prior. In this work, the HDP is used as a comparison method for the new method presented in section 8.

HDP has become very popular in multi-task learning, where multiple data sets are modeled simultaneously. The structure assumed by the HDP is such that it allows modeling multiple data sets with a possibly infinite set of shared components. The probabilities for using the different components can naturally differ for the different data sets. The properties of the HDP are described in more detail in section 8.5.1. Multi-task learning is described in more detail in 5.

4.2.3 Indian Buffet Process

The Indian Buffet Process (IBP) was first presented in [12]. The IBP is a prior for binary matrices with possibly countably infinite columns.

The IBP has properties fundamentally similar to the Dirichlet process: as the Dirichlet process, the IBP involves an infinite sequence of probabilities. The probabilities in the sequence are used for defining the probability of elements of the matrix being 0 or 1.

More formally, the binary IBP matrix $B \sim \text{IBP}(\alpha)$ is generated according to

$$\mu_k \sim \text{Beta}(\alpha, 1) \tag{13}$$

$$\pi_k = \prod_{j=1}^{k} \mu_j \tag{14}$$

$$B(c, k) \sim \text{Bernoulli}(\pi_k) \tag{15}$$

This construction is referred to as the stick-breaking construction for the IBP [13].

The relevance of the IBP to this thesis is through the IBP compound Dirichlet process, which is built on the IBP.

4.2.4 The IBP compound Dirichlet process

One of the recent nonparametric Bayesian priors is the IBP compound Dirichlet process (IBPCD) [27]. The new model presented in section 8 uses a modified IBPCD as a prior for latent structure.

The IBPCD is a prior for a matrix with possibly countably infinite columns containing zeros and values generated from the gamma distribution. The basis of the matrix is an IBP, and the elements with a 1 in the binary matrix are multiplied with a column-specific value from the gamma distribution. The technical details of the IBPCD are presented in section 8.5.2.

The IBPCD aims to remove an assumption in the HDP that is considered problematic; this will be explained in section 8.5.2.

5 Multi-task learning

Multi-task learning is a branch of machine learning in which multiple data sets are modeled simultaneously. The motivation for this is the assumption that by learning the models for the different data sets simultaneously, better performance can be achieved as compared to modeling the data sets separately. This problem has also been called ”learning to learn” and ”transfer learning”. The novel method developed in section 8 is a multi-task model.

An example of a multi-task scenario is a set of data sets comprising of images of people from different cities around the world. Data set 1 could consist of images from Helsinki, data set 2 of images from Rio de Janeiro, and data sets 3-5 of images from Moscow, Stockholm and Bombay correspondingly. Learning the models for different data sets are referred to as *learning tasks*. A learning task could be, for example, to learn to predict the gender of the person in an image from Finland. By taking the multi-task approach we would model the data sets simultaneously in order to have better predictors for new images from the locations as compared to having separate models for each location.

The term *multi-task learning* was first introduced in [3] in the context of neural networks. In [3] predictions for multiple output variables were done based on the same set of input values. The models for making the actual predictions shared submodels for selecting which of the input variables to use. The models also shared information about the parameters used for predictions through shared priors. In [3] the term *task* referred to discriminative prediction tasks. In the last few years, *task* has been used to refer to learning tasks and multi-task learning has been used in general for situations where there are many models, for example, for different data sets, rather than restricting the concept to jointly modeling many discriminative classification tasks.

The multi-task learning approach has been used most often in the *supervised* learning scenario, where the aim of modeling is to predict some quantity of interest, which has been observed in training data. In terms of probability, in supervised learning we are interested in learning to predict Y given X and learn the distribution $p(Y|X)$. The model presented in section 8 takes the multi-task approach in an *unsupervised* setting, where the aim is to build a model for the observed data X and learn $p(X)$ instead of learning a discriminative distribution for some quantity of interest.

The multi-task approach traditionally treats all learning tasks equally: predictive performance in all tasks is considered equally important. Often, as in the case of the model developed in section 8, the interest is, however, asymmetric and the aim of multi-task learning is to improve performance in a particular data set. This is referred to as *asymmetric multi-task learning* and it is described in more detail in section 5.3.

The drawback of multi-task modeling is that modeling many data sets simultaneously is computationally more expensive than simply modeling the data set of interest with a single model (referred to as *single-task learning*). The advantage is that given that the data sets share characteristics, more data will be available for learning the shared parts, allowing generally better predictions. Models for different data sets can share information in very many ways, and recognizing this is essential for increasing predictive performance. If the models for the data sets share parts that are not actually shared in the true generative process, then the extra sharing assumed in the model may actually harm performance.

Therefore the main question in building multi-task models is, how to relate models for different data sets to each other. Quite often the framework of Bayesian hierarchical models is adopted, in which the information in the data used for learning parameter values is shared between different tasks through a shared prior. The idea of hierarchical models is presented in section 3.4.

The optimal way of sharing naturally depends on the structure of the data sets. Various approaches have been taken.

5.1 Multi-task learning by task clustering

One of the well-established ways of relating learning tasks in multi-task learning is by clustering learning tasks into groups and modeling all data within the groups using a shared model. All modeled features withing the clustered data sets are assumed to be similar. The clustering is not known in advance and it is learnt from the data. In terms of probability distributions, the clustering approach assumes that many of the data sets come from the same probability distribution. The clustering approach is taken for example in [14] and [15].

As for the example with data sets of images of people from different locations around the world, the clustering approach would correspond to having a shared model for some combinations of the data sets.

As many gender-related physical characteristics are independent of region (and race), having a shared model for all the data sets might be useful as compared to having data-set-specific models. This approach assumes that all modeled features used in making predictions can be used in the same way for data from different regions.

On the other hand, due to cultural similarity between, for example, Finland and Sweden, clothes worn by men and women in the countries might be more similar to each other than to clothes worn in Rio de Janeiro. Clothes worn in the images might also be useful in predicting gender, and therefore having a shared model for only Finland and Sweden might be more beneficial than having a pooled model for all data, as pooling the data might obscure the cues about gender contained in clothing.

The assumption of many data sets coming from the exactly same distribution is often too strict. Therefore an assumption of a *hierarchy* of the distributions is used for example in [11]. The distributions of the data sets are assumed to come from shared prior distributions. Therefore they will resemble each other and share information without the assumption of their distributions being exactly the same.

5.2 Multi-task learning by modeling shared substructures

Another way of relating learning tasks to each other is to assume that modeled phenomena (and data sets) have shared substructures. This is modeled by including shared components to the models for the different data sets. In terms of probability distributions, this approach assumes that each data set comes from a specific mixture of component distributions, but the overall set of components to choose from is shared.

The difference to the clustering approach is that clustering assumes all features within the clusters to be shared, whereas modeling shared substructures assumes data sets to share some specific set of features. Shared features need to be included to the assumed generative process for the data so as to enable learning them.

An example of this approach with the data sets comprising of images of people from different locations is as follows: many secondary gender-related characteristics are race independent and could be utilized in predictions. For example, upper arms of women are approximately 2 cm longer for a given height. As the difference is subtle, learning to make use of it in predictions might require larger amounts of data than contained in any single data set. Sharing the model for arm length for each gender at different locations while otherwise having separate models for different data sets could help distinguish men and women.

Given that the shared structures exist, by modeling all data sets simultaneously more data will be available for learning the shared substructures. Learning these subparts well will allow better predictions for the data sets in general.

5.3 Asymmetric multi-task learning

Most multi-task models treat all the data sets and models symmetrically, even though often we are only interested in making predictions from one of the models. The models can be constrained by making further assumptions to concentrate their resources on the data set of interest. Learning the model needed to make predictions about the data of interest is referred to as the *task-of-interest* and learning the models for the data sets to be used as background data as *supplementary tasks*.

The task-of-interest is chosen as the distribution of its data is believed to be closest to the distribution of the test set. Future observations are believed to come from this distribution.

Many ways of structuring learning problems in an asymmetric way have been proposed. In [16] increased performance in a task-of-interest is achieved by weighting likelihoods of different data sets differently in optimization. Parameters are fit to better match the data from the task-of-interest and worse performance in training data of supplementary tasks is allowed.

In [17], a sample-specific weighting term is used for samples in a supplementary data set to give samples weights according to their ability to increase performance on training set from task-of-interest.

In [18] all samples from the supplementary tasks are pooled together and each of the background samples is then weighted with a sample specific resampling weight to match the distribution of the pooled background data to that of the task-of-interest.

In [4] and [19] supplementary tasks are assumed to be mixtures of samples from supplementary task-specific distributions and the distribution of the task-of-interest. In terms of probability distributions, data in task-of-interest are modeled as

$$data_{TOI} \sim p(\Theta_{TOI})$$

and data in supplementary task i are modeled as

$$data^{i}_{SUP} \sim \pi_i \cdot p(\Theta_{TOI}) + (1 - \pi_i) \cdot p(\Theta^{i}_{SUP}),$$

where $\pi_i \in [0, 1]$.

The idea presented [4] and [19] is the motivation for the new method presented in section 8. The task-of-interest has less resources than supplementary tasks, in other words the model for the data-set-of-interest is more *sparse* than the models for supplementary data. The model presented in section 8 extends this idea by allowing a probabilistic approach to task-specific sparsity: instead of predefining the number of components in different tasks, we define a distribution which favors having fewer active components in the task-of-interest. The topic of sparsity is explored in section 6.

6 Sparsity

Sparse models refer to models that favor using only few parameters to make predictions and model data in cases where multiple parameters would be available for explaining data. This is the case, for example, with linear regression models, where a linear combination of parameters used to explain quantities of interest could be selected in infinitely many ways when modeling data sets suffering from the small n, large p problem. Instead of using a huge set of parameters with possibly small values, a smaller set of parameters is selected by using some criteria to be used with more significant values and the other parameters are suppressed to zero. In other words, parameter vectors and matrices of sparse models have a lot of zeros.

Assumptions about sparsity are also a way of avoiding overfitting. A model that is allowed an unrestricted set of parameters will have more possibilities of fitting to data as compared to models for which assumptions about sparsity restrict the set of possible parameter combinations and prevent overfitting.

Sparsity has been studied intensively during the recent years. Sparsity has proved itself to be a useful assumption in many applications. Sparse models have been applied, for example, to magnetic resonance imaging data [22] and cell-signaling data from proteomics [23]. Often sparse solutions produce good performance. Sparse solutions also have the advantage of being intuitive: a model that uses few parameters to make predictions is more easy to understand than a model using a huge set of parameters with a negligible value.

The idea of sparsity can be applied at different levels of the model [2]. Imposing sparsity on the parameters of a regression model is an example of applying sparsity at the bottom level of a model. This assumption will suppress most parameters used for regression to zero. An example of applying sparsity to a set of parameters higher in the parameter hierarchy is that of assuming sparsity of the parameters used for controlling clustering in a mixture model. When a mixture model assigns probabilities of samples belonging to different clusters, a sparsity assumption will favor samples having a high probability of belonging to a few clusters and suppress the probability of belonging to the other clusters to zero.

The methods used for promoting sparsity at different levels of model parameters are often, however, the same regardless of the level in the hierarchy at which sparsity is imposed. The most well-known method is L1 regularization. Another method for imposing sparsity is the use of spike-and-slab distributions. The third common approach is to use sparse matrices as building blocks in models. The model developed in section 8 takes this last approach.

6.1 Sparsity by L1 regularization

The most traditional method for inducing sparsity is L1 regularization [24], which enforces most regularized parameters to have value 0.

Model parameters are usually learnt by optimizing them with respect to some cost function. *Regularization* refers to adding terms to the cost function used for optimizing model parameters that penalize the result for the actual values of the parameters with respect to some criteria.

A typical cost function consists of an error term $E(\text{training data}|\theta)$ that describes the training error given parameter values θ and a regularization term $R(\theta)$, where θ denotes a vector of all model parameters. In other words, learning model parameters corresponds to minimizing the cost function with respect to model parameters θ,

$$\text{Minimize}_\theta\{E(\text{training data}|\theta) + R(\theta)\}.$$

The most common type of regularization is the L2 regularization, in which the regularization term

$$R(\theta) = \frac{\lambda}{2} \cdot \theta^T \theta$$

is used. By adjusting the scalar value of λ the level of regularization can be controlled.

In L1 regularization the regularization term is given by

$$R(\theta) = \frac{\lambda}{2} \cdot \sum_{j=1}^{M} |\theta_j|,$$

where M is the dimension of θ.

L1 regularization is easiest to understand by studying the derivatives of the penalties induced by the norm and by comparison to L2 regularization. In L2 regularization the derivative of the penalties for parameter values approaches 0 as the values of the parameters approach 0. In the case of L1 penalty, the derivative does not decrease with parameter values approaching 0. Therefore small parameter values are penalized equally heavily as larger values with respect to their absolute value. This causes the often desired effect of most parameter values being suppressed to 0 and some having a larger value [2].

From the Bayesian view regularization can be seen as a way of incorporating prior information. Different regularizations correspond to different priors for model parameters. For example $L2$ regularization corresponds to a Gaussian prior for model parameters and $L1$ corresponds to a Laplace prior.

6.2 Sparsity by spike-and-slab priors

Another way of implementing sparsity is through *spike-and-slab* priors. Spike-and-slab priors promote sparsity by using mixture distributions for parameter values. Spike-and-slab priors allocate a random variable (parameter of a Bayesian model) x a finite probability of π of having value 0 and divide the remaining probability mass $1 - \pi$ according to some probability distribution. An example of a spike-and-slab prior is

$$p(x) = \pi \cdot \delta(x) + (1 - \pi) \cdot \mathcal{N}(\theta, \sigma),$$

where $\delta(x)$ is the delta function and θ and σ are the parameters of a Gaussian distribution. Spike-and-slab priors were first introduced in [20], more recently they have been discussed for example in [21].

In the model presented in section 8 sparsity is imposed by using sparse matrices, but the solution can be interpreted as a spike-and-slab prior.

6.3 Sparse matrices

Often the sparsity assumption is imposed on a matrix used to control sharing of information. For example, in multi-task learning different learning tasks are assumed to share structure in a sparse manner, which can be interpreted as task-level sparsity.

In practice sparse matrices are used to model the sets of components / variables used by different tasks and observations. For example, rows (denoted by i) of a binary matrix B can correspond to observations and columns (denoted by j) to shared components. $B(i, j) = 1$ would imply that component j is used in modeling observation i.

In [25] a columnwise sparse matrix is used to select variables that are predictive for all tasks. In [26] predictions are done based on a learnt, sparse set of new features, that are linear combinations of original variables.

The aim of imposing task-level sparsity is to enforce tasks to either strongly use some (possibly shared) components for predictions or completely suppress them. The non-sparse alternative is that tasks use very many of the components with possibly negligible weights.

In [27] sparsity is imposed by using binary masking to suppress most components available for explaining the data. This is implemented by multiplying each element of a real-valued matrix of parameters with the corresponding element of a sparse binary matrix. This corresponds to having a *spike and slab* prior for component activities: components are suppressed with some finite probability and given that they are not suppressed, their activity is assumed to have a continuous-valued distribution. This approach is adopted in the model presented in section 8.

7 Topic models

Latent Dirichlet Allocation (LDA) [28] is a well-known generative probabilistic model for collections of count data. LDA models are often referred to as topic models as LDA was originally introduced in the context of text data. The model developed in section 8 has a topic model substructure.

Topic models are easiest to describe by their generative process using text data terminology. Topic models assume that each *document* (observation) deals with a finite set of *topics* with corresponding probabilities. Each observed *word* in the document is generated from one of the document specific topics. The *vocabulary* of the data set is the set of different word identities observed in the data. The word identities are referred to as *terms*. Topics are shared across documents and each topic has a specific topic-to-word (or more accurately topic-to-term) distribution, which defines the probabilities for generating different terms.

Actual word counts are observed. Topics are latent variables, and they are learnt from the data to resemble sets of words that often occur together.

The formal generative process for a topic model, as first presented in [28], is:

1. Draw $N \sim \text{Poisson}(\xi)$.

2. Draw $\theta \sim \text{Dirichlet}(\alpha)$, where the dimensionality of θ is known to be K,
which corresponds to the number of topics.

3. For each of the N words w_n

(a) Draw topic index $z_n \sim \text{Multinomial}(\theta)$
(b) Draw word identity for word w_n ($w_n = term_i$, e.g. $w_n =$ “cat”) from $p(w_n|z_n, \beta)$,
a multinomial probability conditioned on the topic z_n, where β contains the
parameters for different topics.

Hyperparameter ξ, parameter vector α and parameter β have the same values for all documents.

The generative process is repeated for each document in the data. This generative process assumes that word probabilities are not assigned hierarchical priors but are estimated as parameters, which leads to simple equations. This generative process does not use the assumption about the distribution of the number of words in a document for anything, and it is as such a redundant assumption.

Figure 1 contains the *plate diagram* for topic models. A plate diagram describes the relationships between the parameters of a generative model by presenting them in a Bayesian network. Random variables (parameters of the model) are the nodes of the network. In a Bayesian network, the distribution of a random variable is known when the values of all the parameters, from which an arrow is drawn to the random variable of interest, are known. The values of variables presented by the grey nodes are known whereas values of parameters corresponding white nodes are inferred from the data. Plates denote that the random variables within them are generated multiple times (usually denoted by the character in the corner of the plate).

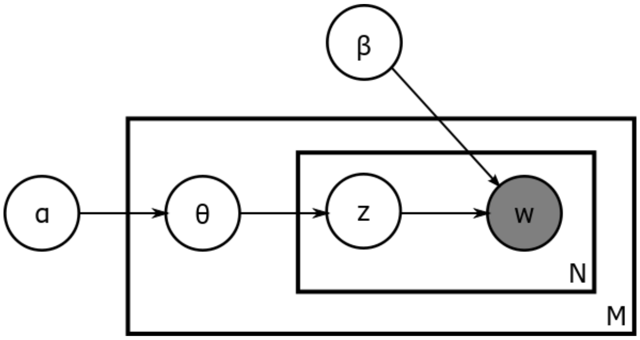

Figure 1: Plate diagram for latent Dirichlet allocation model (topic model). M denotes the number of documents and N the number of words in a document.

The difference between topic models and the more traditional family of mixture models is as follows: mixture models assume all variables of samples to be generated by one mixture component, even though the exact identity of that component is not known. Topic models assume that the variables of a sample are explained by a set of topics that are simultaneously active in the sample.

As an example of the difference between topic and mixture models, a mixture modeling approach to modeling observed symptoms of a patient would correspond to assuming that all symptoms are caused by the same disease, but we do not know which particular disease. A topic modeling approach would correspond to assuming that a set of diseases cause the (possibly overlapping) symptoms. A mixture model could be used to compute posterior probabilities for the patient having different diseases whereas the topic model could be used to compute how strongly different diseases are present in the patient.

Topic models have been applied widely to gene expression data (for example [29], [30] and [31]). In these cases the data has been preprocessed in such a way that the originally real-valued measurements are converted into counts. The motivation for the model presented in section 8 comes from biology, and as topic models have been found successful in modeling gene expression data, using topic models is a natural choice.

8 A novel approach to small n, large p: Shielded models

We propose a new multi-task model to alleviate the small n, large p problem. Our model builds on the model family of topic models and uses a nonparametric Bayesian prior to avoid the inconvenience of selecting model complexity within our chosen model family. The novelty of the model is that asymmetric sparsity is used to focus the optimization of model performance asymmetrically across tasks, more precisely to promote performance in a task-of-interest.

The motivation for our model stems from biology and more specifically from modeling simultaneously multiple species. Different species can be assumed to share biological processes at the cellular level. Even though the phenotypes of the species differ strongly, even distantly related species share huge portions of their genome. More specifically, important biological pathways are likely to be relatively conserved, as their function can't be jeopardized at any step of the way in the evolutionary process. Therefore it is rational to assume that for example man and mice share some biological processes at the cellular level.

Usually our interest is, however, asymmetric. We are not interested in the performance of our model on most of the data sets, their purpose is mainly to augment the data in a task-of-interest. This thesis studies targeting model performance by constraining model complexity for different data sets unevenly.

The model developed is sparse. The variation in data is assumed to be explained by a set of components, which are either relatively active or completely suppressed in the different tasks. By making this assumption we wish to constrain the results and make them better interpretable.

The sparsity of the task-of-interest is constrained more than that of supplementary tasks. There are two important motivations for doing this. First, by constraining the resources of the task-of-interest, it will be forced to share components with the supplementary tasks. The second motivation stems from knowing that supplementary tasks can contain features that are not present in task-of-interest: the supplementary tasks need to have resources to explain these away so that they do not need to use the resources shared with the task-of-interest for modeling the unshared features. Therefore supplementary tasks are allocated more resources than the task-of-interest to *shield* the shared features. Models implementing asymmetricity by allocating extra resources for supplementary tasks are referred to as *shielded models.*

8.1 Hypothesis about the effects of asymmetric sparsity

We assume that allocating the task-of-interest less resources than the supplementary tasks will promote performance in the task-of-interest. If the task-of-interest is allocated too few resources, performance in the task-of-interest will naturally decrease.

8.2 Technical description of the generative process

We model C data sets indexed by c simultaneously with a model that falls within the framework of topic models. Each data set has D_c documents (observations), documents are indexed by d. Each document has $N_{d,c}$ words that are indexed by n. Each document consists of words $w_{c,d,n}$. The term *dictionary length* refers to the number of different terms in the corpus, in other words, the dimensionality of the data. Each word is generated from one of the numerous topics indexed by k. The topic of word $w_{c,d,n}$ is denoted by $z_{c,d,n}$.

Each topic k has a specific topic-to-word distribution, which defines the probabilities of generating different terms, when topic k is observed. The topic-to-word distributions are multinomial distributions over the finite vocabulary, $w_{c,d,n}|z_{c,d,n} = k \sim \text{Multinomial}(\boldsymbol{\beta}_k)$. The number of terms in the dictionary is referred to as the dictionary length. Parameters $\boldsymbol{\beta}$ are distributed according to $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\eta)$.

Within each task (data set), each document has a document-specific *topic distribution*, which defines the probability of different topics generating words in that particular document. The topic distribution is a multinomial distribution with a document-specific parameter $\theta_{d,c}$, which on its behalf is distributed according to the Dirichlet distribution, $p(z_{c,d,n}) \sim \text{multinomial}(\theta_{d,c})$ and $\theta_{d,c} \sim \text{Dirichlet}(\omega_c)$.

Documents within each task are assumed to have similar topic distributions. This is modeled by assuming that document-specific topic distribution parameters follow a task-specific Dirichlet distribution, in other words $\theta_{d,c} \sim \text{Dirichlet}(\omega_{\mathbf{c}})$.

Also the total number of words in each task $n_c^{(\cdot)}$ is modeled as a function of ω_c: $n_c^{(\cdot)} \sim \text{NB}(\sum_k \omega_c^k, \frac{1}{2})$. This assumes that the number of words in a task depends on the number (and strength) of topics active in that task. Inversely, the number of active topics in a task with few words (data) is lower as compared to a task with lots of words.

We assume that the number of topics is in principle countably infinite, but for modeling any finite data set, only a finite subset is used. This is achieved by generating the parameters $\theta_{d,c}$ from a prior which does not specify a limit for the number of possibly active topics. The prior distribution is a Dirichlet distribution over a set of infinite topics, but only a finite subset of the topics is given a probability that differs from 0. The finiteness of the set of active topics is controlled by constructing the parameter ω_c as a product of elements form appropriate distributions.

The parameter $\omega_{\mathbf{c}}$ is a Hadamard product of three different parameters, which all have a different role in the process, $\omega_c = \mathbf{b}_c.\boldsymbol{\phi}_c.\boldsymbol{\psi}_c$.

The parameter $\mathbf{b}_c$ is a row of an IBP matrix, in other words $\mathbf{B} \sim \text{IBP}(\alpha)$ and $\mathbf{b}_c = \mathbf{B}(c, :)$ following the Matlab notation; here alpha is the base parameter of the IBP. This allows having a possibly countably infinite number of topics and learning the number of topics from the data. The parameter $\mathbf{b}_c$ is binary, topics k for which $\mathbf{b}_c^k = 0$ will not be used in task c.

The stick-breaking construction of the IBP is used, and in addition to the binary matrix $\mathbf{B}$, topic specific stick parameters π_k are generated.

The parameter $\boldsymbol{\phi}_c$ is used to model topic strengths within task c. The k^{th} element of $\boldsymbol{\phi}_c$, $\phi_c^{(k)}$, corresponds to the strength of topic k in task c and a prevailing topic will have a larger $\phi_c^{(k)}$ as compared to other, weaker topics. Topic strengths in different tasks are assumed to be related. This is modeled by assuming that $\phi_c^{(k)} \sim \text{Gamma}(\gamma^{(k)}, 1)$, where $\gamma^{(k)}$ is a topic specific shape parameter, which is shared for all tasks. The *scale* formulation of the gamma distribution is used and 1 is the scale of the distribution. The parameter $\gamma^{(k)}$ is distributed according to $\gamma^{(k)} \sim \text{Gamma}(a_1, a_2)$, where $a_1 = a_2 = 1$.

The parameter $\boldsymbol{\psi}_c$ is a task-specific binary vector used to impose excess sparsity. Elements $\psi_c^{(k)}$ of $\boldsymbol{\psi}_c$ are assumed to be generated from a Bernoulli distribution, $\psi_c^{(k)} \sim \text{Bernoulli}(\epsilon_c)$, where ϵ_c is a task-specific parameter. Multiplication with $\boldsymbol{\psi}_c$ (referred to as *binary masking*) will suppress some of the elements of $\mathbf{b}_c.\boldsymbol{\phi}_c$ to zero, and by varying the parameter ϵ_c the level of this suppression can be controlled.

Section 8.3 describes the formal generative process of the model.

8.2.1 Multiple uses of the binary masking

The binary masking can be used to impose asymmetric sparsity: when tasks c have a different value of the parameter ϵ_c, the penalties for activation of topics in the different tasks will differ and the tasks will probably use a different number of topics.

However, the binary masking can also be used to impose a general resource constraint: by making $\epsilon_c < 1$ for all tasks, the number of topics activated in each task will be reduced.

These two functions of the binary masking are not mutually exclusive: asymmetric sparsity can be imposed simultaneously with a general resource constraint.

8.3 The formal generative process

This section describes algoritmically how to generate data from the new model. Tasks are indexed using c, documents using d and words using n. The generative process for shielded multi-task topic model is:

1. Draw a binary matrix $\mathbf{B} \sim \text{IBP}(\alpha)$ and parameters π_k

    (a) $\mu_k \sim \text{Beta}(\alpha, 1)$

    (b) $\pi_k = \prod_{j=1}^{k} \mu_j$

    (c) $b_c^{(k)} \sim \text{Bernoulli}(\pi_k)$ for each c

2. For each component $k = 1, 2, ...$

    (a) Draw $\gamma^{(k)} \sim \text{Gamma}(a_1, a_2)$

    (b) Draw the topic distribution $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\eta)$

3. For each task $c = 1, 2, \ldots, C$

    (a) For each component $k = 1, 2, ...$

(a) Draw topic strengths $\phi_c^{(k)} \sim \text{Gamma}(\gamma^{(k)}, 1)$
(b) Draw $\psi_c^{(k)} \sim \text{Bernoulli}(\epsilon_c)$

(b) For every document $d = 1, 2, ...D_c$ in task c

(a) Draw the distribution over topics
$\boldsymbol{\theta}_{c,d} \sim \text{Dirichlet}(\mathbf{b}_c \cdot \boldsymbol{\phi}_c \cdot \boldsymbol{\psi}_c)$

(b) For each word $n = 1, 2, ..., N_{d,c}$ in the document

(a) Draw the topic index $z_{c,d,n} \sim \text{Multinomial}(\boldsymbol{\theta}_{c,d})$
(b) Draw the term $w_{c,d,n} \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{c,d,n}})$

4. To specify a full generative model we also model the size of a task:

$$n_c^{(\cdot)} \sim \text{NB}(\textstyle\sum_k b_c^{(k)} \phi_c^{(k)} \psi_c^{(k)}, \frac{1}{2})$$

Figure 2 contains the plate diagram for the model.

8.4 Posterior computation

A mixture of Gibbs sampling and the Metropolis-Hastings algorithm is used to sample from the posterior distribution of the model parameters.

8.4.1 Sampling z

Topic assignments $z_{c,d,n}$ are sampled using Gibbs sampling described in section 3.3.4.

In order to sample $z_{c,d,n}$, we integrate the probability of observing topic k over the set of possible underlying topic distributions $p(z_{c,d,n}) \sim \text{multinomial}(\theta_{c,d})$. The posterior for the topic assignment of the n^{th} word in document d of task c is

$$\begin{aligned} &p(z_{c,d,n} = k | \mathbf{z}_{\setminus c,d,n}, w_{c,d,n}, \Delta) \propto \\ &(n^{(k)}_{w_{c,d,n}, \setminus c,d,n} + \eta) \int p(z_{c,d,n} | \boldsymbol{\theta}_{c,d}) p(\boldsymbol{\theta}_{c,d} | \mathbf{z}_{\setminus c,d,n}, \Delta) \; d\boldsymbol{\theta}_{c,d} \end{aligned} \tag{16}$$

where $\Delta = \{\boldsymbol{\phi}_c^\bullet, \boldsymbol{\pi}_c^\bullet, \boldsymbol{\gamma}, \alpha, \epsilon_c\}$. The symbol $\mathbf{z_{\setminus c,d,n}}$ denotes the current topic assignments for all other words in the data (assignments for all other words except for the word $w_{c,d,n}$). Parameters $\boldsymbol{\phi}_c$ and $\boldsymbol{\pi}_c$ are infinite-dimensional. Their values are known for the topics k which have appeared in the data ($z_{c,d,n} = k$ for some word $w_{c,d,n}$). The known values are denoted by the closed ball superscript $^\bullet$, whereas the values of the parameters for the currently unobserved topics are denoted by the open ball superscript $^\circ$. When referring to all elements of the parameters, superscript ^{all} is used.

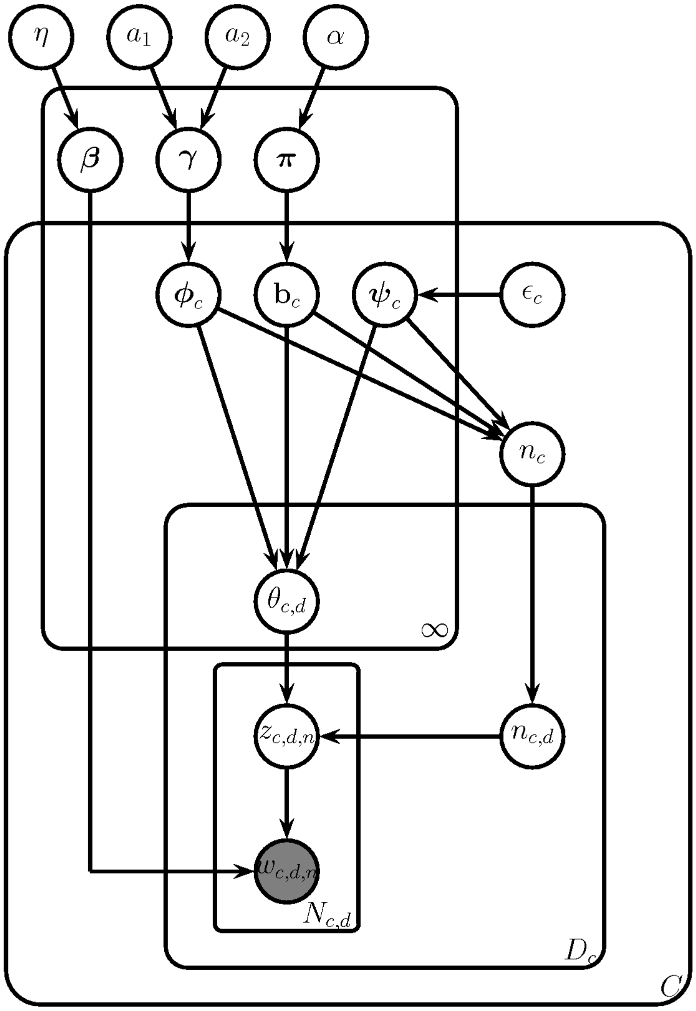

Figure 2: Plate diagram for shielded multi-task topic model. Topic indexes are omitted for visual clarity. Variables n_d and n_c are auxiliary variables to technically present the flow of information. The number of words in data set c is modeled as a function of $\mathbf{b}_c \cdot \boldsymbol{\phi}_c \cdot \boldsymbol{\psi}_c$, and variables n_d and $n_{c,d}$ deterministically denote the numbers of words generated from different topics in the documents and tasks, respectively.

Equation 16 involves an integration over the sparse IBP matrix, which is combinatorial:

$$p(\boldsymbol{\theta}_{c,d}|\mathbf{z}_{\backslash c,d,n}, \Delta) \propto$$
$$\int \sum_{\mathbf{b}_c} \sum_{\boldsymbol{\psi}_c} p(\boldsymbol{\theta}_{c,d}|\boldsymbol{\psi}_c, \mathbf{b}_c, \boldsymbol{\phi}_c, \mathbf{z}_{\backslash c,d,n}) p(\mathbf{b}_c, \boldsymbol{\psi}_c, \boldsymbol{\phi}^\circ|\boldsymbol{\phi}_c^\bullet, \boldsymbol{\pi}^\bullet, \boldsymbol{\gamma}, \alpha, \epsilon_c)\ d\boldsymbol{\phi}_c^\circ \qquad (17)$$

In order to evaluate 16, it is, however, sufficient to be able to compute the integral for each topic k one at a time. Therefore it is also sufficient to compute the probabilities given by equation 17 for each topic at a time.

The integral

$$\int p(z_{c,d,n}|\boldsymbol{\theta}_{c,d})p(\boldsymbol{\theta}_{c,d}|\mathbf{z}_{\backslash c,d,n},\Delta)\ d\boldsymbol{\theta}_{c,d} = E[\boldsymbol{\theta}_{c,d}|\mathbf{z}_{\backslash c,d,n},\Delta] \tag{18}$$

can be evaluated for each element of $\theta^{(k)}_{c,d}$ separately. The corresponding expectation is approximated in Appendix A.

8.4.2 Sampling the stick parameter

The IBP stick parameters π^k for the active topics are needed for sampling topic assignments, as shown in Appendix A. Active topics are topics that have generated words, in other words topics k for which $n^k_{(.)(.)} > 0$

To sample the stick parameters for the active topics, we follow the semi ordered stick breaking scheme detailed in [13]. The stick parameters for the active topics are distributed according to:

$$p(\pi^{\bullet}_k|\mathbf{B}) \sim \text{Beta}\left(\sum_{c=1}^{C} b^{(k)}_c, 1 + C - \sum_{c=1}^{C} b^{(k)}_c\right) \tag{19}$$

where $\mathbf{B}$ is the current value of the IBP (the binary matrix). The posterior can be sampled directly using Gibbs sampling.

A topic is inactive if it does not appear anywhere in the whole corpus, in other words $n^k_{(.)(.)} = 0$ even if $\sum_c b^{(k)}_c > 0$, and active otherwise. The inactive topics have an ordering of decreasing stick lengths:

$$\begin{aligned} &P(\pi^{\circ}_k|\pi^{\circ}_{k-1}, \mathbf{z}_{k:k>K^{\dagger}} = 0) \propto \\ &\quad \exp\left(\sum_{i=1}^{N}\frac{1}{i}(1-\pi^{\circ}_k)^i\right)\pi^{\circ}_k(1-\pi^{\circ}_k).\mathbb{I}(0 \leq \pi^{\circ}_k \leq \pi^{\circ}_{k-1}). \end{aligned} \tag{20}$$

Stick parameters for the inactive topics are sampled using equation 20 by *adaptive rejection sampling* (ARS) [32]. ARS samples from a distribution $p(x)$ by first constructing an envelope function for $\log(p(x))$. The envelope function is then used for rejection sampling. Whenever a sample is rejected, the envelope function is updated to correspond better to the underlying density. The R package 'ars' [33] is used to generate samples using ARS.

8.4.3 Reinstantiating the IBP matrix B and Bernoulli masking matrix $\boldsymbol{\Psi}$

Even though topic assignments can be sampled while integrating over the binary IBP matrix, the IBP matrix is still needed for sampling the stick parameters for the active topics. Therefore after sampling topic assignments z, the current value of the binary matrix $\mathbf{B}$ (actual value of the IBP) needs to be reinstantiated.

The current value of the IBP matrix is reinstantiated according to

$$p(b_c^{(k)} = 1|\pi_k, \boldsymbol{\phi}_k, \psi_c^{(k)}, \mathbf{z}_c) = \begin{cases} 1, & \text{if } n_{c,(.)}^{(k)} > 0 \\ \pi_k, & \text{if } n_{c,(.)}^{(k)} = 0, \psi_c^{(k)} = 0 \\ \frac{\pi^{(k)}}{\pi^{(k)}+2^{\phi_c^{(k)}}(1-\pi^{(k)})}, & \text{if } n_{c,(.)}^{(k)} = 0, \psi_c^{(k)} = 1 \end{cases} \tag{21}$$

where $\phi_c^{(.)} = \sum_k \phi_c^{(k)} \cdot b_c^{(k)} \cdot \psi_c^{(k)}$. In the first case, topic k has generated words in task c. In the second case, topic k has not been observed in task c and parameter $\psi_c^{(k)} = 0$ prevents the topic from being active regardless of the value of $b_c^{(k)}$. In the third case, topic k has not been observed in task c but it could be active as $\psi_c^{(k)} = 1$.

The masking vector $\boldsymbol{\psi}_c$ can be reinstantiated in a similar way:

$$p(\Psi_c^{(k)} = 1|\epsilon_k, \boldsymbol{\phi}_k, \psi_c^{(k)}, \mathbf{z}_c) = \begin{cases} 1: & \text{if } n_{c,(.)}^{(k)} > 0 \\ \epsilon_k: & \text{if } n_{c,(.)}^{(k)} = 0, b_c^{(k)} = 0 \\ \frac{\epsilon^{(k)}}{\epsilon^{(k)}+2^{\phi_c^{(k)}}(1-\epsilon^{(k)})} & \text{if } n_{c,(.)}^{(k)} = 0, b_c^{(k)} = 1 \end{cases} \tag{22}$$

8.4.4 Sampling topic strength parameters $\phi_c^{(k)}$ and $\gamma^{(k)}$

Topic strength parameters $\phi_c^{(k)}$ are sampled using the Metropolis-Hastings algorithm described in section 3.3.3.

The joint probability of $\phi_c^{(k)}$ and total number of counts assigned to topic k can be expressed as

$$p(\phi_c^{(k)}, n_c^{(k)}|\gamma^{(k)}, b_c^{(k)}, \psi_c^{(k)}) = \frac{(\phi_c^{(k)})^{\gamma^{(k)}-1}\, e^{-\phi_c^{(k)}}}{\Gamma(\gamma^{(k)})} \prod_{c:b_c^{(k)}.\psi_c^{(k)}=1}^{C} \frac{\Gamma(n_c^{(k)} + \phi_c^{(k)})}{\Gamma(\phi_c^{(k)})\, n_c^{(k)}!\, 2^{(\phi_c^{(k)}+n_c^{(k)})}}. \tag{23}$$

We use the Metropolis Hastings algorithm and equation 23 to generate samples from the posterior of $\phi_c^{(k)}$.

Also samples from the posterior of $\gamma^{(k)}$ can be generated by using the joint distribution

$$p(\gamma^{(k)}, \phi_{(.)}^{(k)}|n_c^{(k)}, b_c^{(k)}, \psi_c^{(k)}, a_1, a_2) = p(\gamma^{(k)}|a_1, a_2) \prod_{c:b_c^{(k)}.\psi_c^{(k)}=1}^{C} p(\phi_c^{(k)}|n_c^{(k)}\gamma^{(k)}, b_c^{(k)}, \psi_c^{(k)}) \tag{24}$$

and the Metropolis Hastings -algorithm.

8.5 Comparison to earlier work

The most similar existing methods are the *Hierarchical Dirichlet Process* multi-task model [11] and the IBP Compound Dirichlet Process [27] single-task model.

8.5.1 Hierarchical Dirichlet Process multi-task model

The Hierarchical Dirichlet Process multi-task model (HDP) is a nonparametric Bayesian model that generates topics from a Hierarchical Dirichlet Process to be shared in different tasks. The number of topics is not chosen but learnt from data as with our model. HDP provides a model for similar problem setups as our approach, whereas some important assumptions made by the HDP are different.

HDP allocates a similar amount of resources for all tasks. The major contribution of this thesis is the study of the effects related to altering task-specific resources to improve performance in the task-of-interest.

HDP does not favor sparse solutions, that is, it allows topics with very small prevalences. Our model is sparse and it favors solutions with a smaller number of topics with high prevalence.

HDP first generates a possibly infinite set of topics with associated probabilities for their prevalence in tasks. The probability for using the topics in documents within tasks is a corrupted version of the task-level probabilities for the topic usage. Therefore topic prevalence over tasks is positively correlated with topic prevalence in documents within the tasks. This assumption can naturally be very far from the true structure of the data; it is easy to imagine a topic that appears often but generates only few words. For example, in a corpus containing documents describing cars from different manufacturers, the topic "brakes" will probably appear in almost all documents (task prevalence close to 1), but most words in the documents will probably not deal with the topic (prevalence in documents is substantially lower than 1).

Figure 8.5.1 contains the plate diagram for the HDP multi-task model. The generative process for the HDP multi-task model is as folows:

1. Draw $G_0 \sim \mathrm{DP}(H, \gamma)$

2. For tasks $c = 1, \ldots, M$

    (a) Draw $G_c \sim \mathrm{DP}(G_0, \alpha_0)$

    (b) For every document $d = 1, \ldots, D_c$ in task c

        (a) Draw $G_{c,d} \sim \mathrm{DP}(G_c, \alpha_1)$

        (b) For every word n in document d

            (a) Draw topic $z_{c,d,n}$ from Dirichlet$(G_{c,d})$

            (b) Draw word $w_{c,d,n}$ from Multinomial$(\boldsymbol{\beta}_{z_{c,d,n}})$

where M denotes the number of tasks (data sets) indexed with c. Each task c has D_c documents. The number of words in a document is $N_{c,d}$. Parameters $\boldsymbol{\beta}_{z_{c,d,n}}$ for topic-to-word distributions are generated from $\boldsymbol{\beta}_k \sim Dirichlet(\eta)$ as new topics are generated. H, γ, α_0 and α_1 are the hyperparameters of the model.

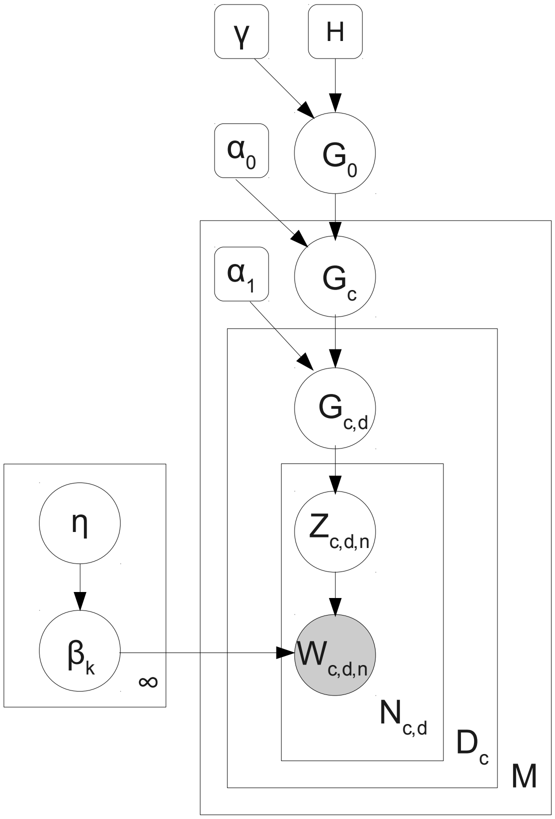

Figure 3: Plate diagram for the hierarchical Dirichlet multi-task model.

8.5.2 Focused topic models

Focused topic model (FTM) [27] is a single-task model that has a structure very similar to our model. FTM models a set of related documents sharing a possibly infinite set of topics. FTM uses an Indian Buffet Process to generate a binary matrix to control sharing of topics. Topic prevalence (the probability of a topic appearing in documents) is decoupled from topic prevalence within the documents. This is achieved by modeling topic prevalence with a gamma distribution that is independent from the IBP controlling topic sharing.

Our model uses a similar structure for decorrelating task-level sharing and document-level prevalence. This structure also allows a natural way of implementing the novel asymmetric resource allocation. We extend the model structure used in FTM by an additional independent binary masking to shut down topics in tasks with task-specific probabilities to promote performance in task-of-interest.

FTM models a set of related documents. We model a set of related learning tasks, within which all documents are assumed to use the same set of topics. This extends the single task approach of FTM to a multi-task scenario.

Focused topic models connect the parameters controlling topic prevalence within documents through a shared prior. In other words, all topics are assumed to occur equally often. The distribution used for achieving this is, however, relatively loose, and the assumption is a weak one. Our model assumes that topic strengths are similar in different tasks, in other words that the strength of topic k is similar in all tasks.

Figure 8.5.2 presents a plate diagram for the focused topic models. The generative process for a focused topic model for a data set with M documents (indexed by m) is as follows

1. For topics $k = 1, \ldots$

   (a) Draw π_k according to equation 9.7

   (b) Draw relative weight of topic $\phi_k \sim \text{Gamma}(\gamma, 1)$

   (c) Draw parameter of the topic-to-word distribution $\beta_k \sim \text{Dirichlet}(\eta, 1)$

2. For documents $m = 1, \ldots, M$

   (a) Draw a row of and IBP matrix $\mathbf{b_m}$ according to equation 9.7

   (b) Draw the total number of words $n^m_\cdot$ in document m,
   $n^m_\cdot \sim \text{NegBin}(\sum_k (b_{m,k}\phi_k), \frac{1}{2x})$

   (c) Draw the distribution over topics $\theta_m \sim \text{Dirichlet}(\mathbf{b_m} \cdot \phi)$,
   where ϕ is a vector of weights ϕ_k.

   (d) For each word $n = 1, \ldots, n^m_\cdot$

      (a) Draw a topic $z_{m,n} \sim \text{Dirichlet}(\theta_m)$

      (b) Draw word $w_{m,n} \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{m,n}})$.

The most similar existing multi-task method (HDP) is neither sparse nor asymmetric and it differs technically in most aspects from our approach. Technically the most similar existing method (IBPCD) is designed for very different learning scenarios than our model.

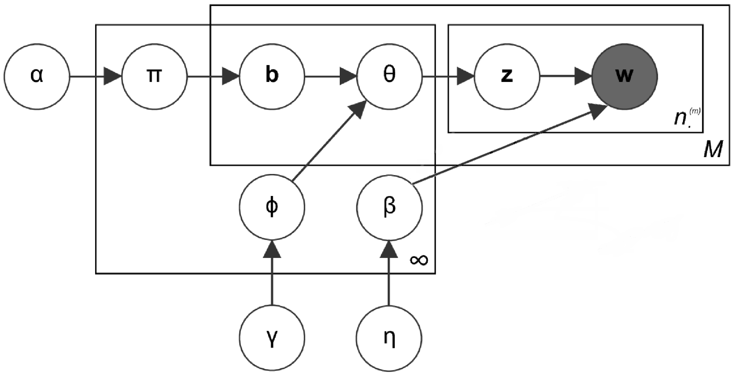

Figure 4: Plate diagram for FTM.

9 Experiments

A set of experiments was conducted to study the effect of shielding and to compare the performance of the developed model to the existing state-of-the-art method. The performance of a model is used as a measure for the goodness of the approach.

The new method was compared to the existing state-of-the-art method on real data and in addition experiments were conducted with toy data. Data sets used in the experiments are described in section 9.3.

9.1 Measuring performance

The performance of our model and the comparison method was evaluated by computing predictive likelihoods, $\int_\Theta p(\text{test data}|\Theta) \cdot p(\Theta|data, prior)$, for test data. Test data consist of the bag-of-words representations of the test documents, in other words, how often each term in the vocabulary has been observed in the test documents.

Computing predictive likelihoods for the new model and the comparison methods includes integration over the document-specific topic distributions of the documents in the test set, as they are not known for the previously unobserved documents of the test set. A numerical approach referred to as *empirical likelihoods* is taken to avoid solving the integral analytically and to produce the estimates of the predictive likelihoods.

For toy data the true distribution of the test set is known, and therefore predictions given by our model can be directly compared to the optimal predictions obtained by using the true model. Toy data results are reported as the difference of logarithmic likelihoods computed using our model and the true model. The difference is normalized by the word count of the test data to give comprehensible average word results:

$$\text{difference to the true model} = \frac{\log\left(\dfrac{p(\text{test data}|\text{learnt parameters})}{p(\text{test data}|\text{true parameters})}\right)}{\text{number of words in test set}} \tag{25}$$

For example, a reported logarithmic difference of -0.15 means that the geometric mean of the likelihoods of the words in the test set computed by using learnt parameters is $\exp(-0.15) = 0.8607$ of the optimal predictive likelihoods.

For real data, predictive likelihoods are converted into *perplexities* that is the standard measure of performance in topic models literature. The perplexity of test data consisting of words $w_1, \ldots, w_{n_T}$ is computed according to

$$\text{Perplexity}(w_1, \ldots, w_{n_T}) = \exp\left(-\frac{1}{n_T}\log(p(w_1, \ldots, w_{n_T}|\text{training data}))\right) \tag{26}$$

where n_T is the number of words in the test data. The dictionary (the set of different terms that appear in documents) is naturally assumed to be the same as in the training set.

Perplexity can be interpreted as the "confusedness" of the model. The interpretation of perplexity is easiest to describe by an example. A test data perplexity of 10 can be interpreted as follows: the model is as confused by each word in the test data as if it had to choose them uniformly from a set of 10 alternatives. The simplest of models which assigns an equal probability for all words in the dictionary would therefore have a perplexity of the number of terms in the dictionary.

9.2 Empirical likelihoods

The method of empirical likelihoods can be used to numerically estimate predictive likelihoods when the computation of predictive likelihoods includes some intractable integrals. The method of empirical likelihoods is described, for example, in [34].

A slightly modified version of the method of empirical likelihoods is used to estimate the predictive likelihoods in the experiments. The original aim was to use the standard method, but unfortunately the description of the method was misinterpreted. At the time of the discovery of the mistake, there was not enough time to correct it. The modified version of the method was, however, used consistently for our model and the comparison method, and it has a sensible interpretation as the lower bound of the classical method. In addition, the comparison method results on the NIPS data set obtained by using the modified method do not differ observably from the results in [11] obtained with the classical method as presented. The difference between the methods is discussed in more detail in Appendix B.

An approximation for the predictive likelihoods is computed as follows:

1. For each posterior sample s

   (a) Generate N_{emp} *sample topic distribution parameters* θ^s indexed by v as θ^s_v, $v = 1, ..., N_{emp}$ from $\theta^s \sim \text{Dirichlet}(\mathbf{b}^s_c.\boldsymbol{\phi}^s_c.\boldsymbol{\psi}^s_c)$, where $\mathbf{b^s_c}$, $\boldsymbol{\phi}^s_c$ and $\boldsymbol{\psi}^s_c$ are the values of the corresponding parameters in posterior sample s

   (b) Compute the average log likelihood for different terms i in the dictionary: this average log-likelihood is an approximation to the empirical log-likelihood given by posterior sample s to a single occurrence of term i. The approximation is

$$EL_s(term_i) = \frac{\sum_{v=1}^{N_{emp}} \log(p(\text{word} = term_i|\theta^s_v, \beta^s))}{N_{emp}}$$

where β^s contains the learnt parameters for the topic-to-word distributions in posterior sample s.

For the HDP models, the process is similar. The only difference is that topic distribution parameters are generated from a different distribution, as defined in section 8.5.1.

To compute the modified empirical likelihoods for a test data containing documents $d = 1, \ldots, D_{test}$ where each document has words $w = 1, \ldots, N_{words,d}$, the average over the observed word counts, test documents and posterior samples s is taken,

$$EL(\text{test data}) = \frac{\sum_{s=1}^{S}\sum_{documents}\sum_{w=1}^{N_{words,d}}\sum_{i=1}^{N_{terms}} \mathbb{I}(\text{word}_{d,w} = term_i)EL_s(term_i)}{S \cdot \sum_{dindocuments} N_{words,d}}, \tag{27}$$

where S is the number of posterior samples and N_{terms} is the number of terms in the dictionary.

9.3 Data sets

We perform four experiments on toy data and one on real data. All experiments using toy data generate it as described in section 9.3.1. The experiment with real data is done using the NIPS data set described in section 9.3.2.

9.3.1 Toy data

Experiments with toy data were used to study the shielding effect and to study properties of the new model. In these experiments, the new model was not compared to comparison methods.

Toy data was generated from a distribution that belongs to the model family of the developed model. Some parts of the distribution were, however, fixed to facilitate computation and interpretation of results.

The generation of toy data follows the major guidelines of the generative process presented in section 8.3. Instead of following the random process defined by the model to generate latent structure contained in $\Phi, \mathbf{B}$ and $\boldsymbol{\psi}$, we fixed such a structure (the values of $\mathbf{B}$ and $\boldsymbol{\psi}$) in which the task-of-interest has 2 topics, and each supplementary task shares one of them. In addition to the shared topic, each supplementary task has a task-specific topic, which is not shared with any other task. The value of parameter ϕ was set to 300 for all tasks and topics. The values of the hyperparameters for $\Phi, \mathbf{B}$ and $\boldsymbol{\psi}$ do not have any impact on the data after setting the values of this set of parameters.

Topic-to-word distributions were generated according to the *sparse topic models* presented in [35]. The probability of generating words in the corpus comes from a Dirichlet distribution. Unlike in the assumptions of our model, sparse topic models allocate a different pseudocount for different words. The pseudocounts were fixed in such a way that each topic has a set of words, for which it has the highest pseudocount (0.5). For other words the pseudocount is lower(0.03). Each topic had approximately equally many words for which the pseudocount was higher. *Sparse topic models* -structure was used instead of the generative process assumed in the new model to generate data that is easy to learn. This allowed doing experiments in which the amount of training data is small and sampling is correspondingly computationally inexpensive.

The number of supplementary tasks was fixed to 9 in all experiments with toy data. The number of terms in the dictionary (*dictionary length*), the number of documents in the task-of-interest and supplementary tasks and the number of words per document were varied in different experiments.

For toy data, the test set consists of 2000 test documents with dictionary length $\cdot$ 1000 words in each document generated from the model.

An important aspect of the toy data is that the true number of components is the same in the task-of-interest and the supplementary tasks. The sparsity assumption does not as such match the toy data.

9.3.2 NIPS data

We used the NIPS data set to compare the performance of the new model with a state-of-the-art method (the HDP multi-task model).

The NIPS data set[1] consists of NIPS articles from 1987 to 1999. The articles deal with a range of topics spanning nine sections. Standard stop words and words occuring more than 4000 times or fewer than 50 times in the whole corpus have been removed. We select the five most frequent sections as learning tasks: Neuroscience (NS), Learning Theory (LT), Algorithms and Architecture (AA), Applications (AP) & Control, Navigation and Planning (CN). The resulting data set contains 1147 articles, the vocabulary size is 1321 and the average document length is ~ 950 words.

To compare the new model developed in this thesis to existing state-of-the-art methods on real data, cross validation is used to assess the performances of the models. The data set is divided into 5 training sets and a validation set. Unlike in the cross validation process described in section 4.1, the roles of the subsets are not changed: parameters of different models are always fitted to one of the training set subsets and performances are always validated on the same test set. The number of training samples in the task-of-interest was varied ($5, 10, 20$ and 40) and the number of documents in supplementary tasks was 10 or 25. The number of documents in the test set is 16.

9.4 Experiment 1: Fixing hyperparameter α

The aim of the first experiment was to study the shielding effect and the effect of the hyperparameter α. The model was run with $\alpha = 3, 9$ and 15.

The effect of asymmetric sparsity was explored using the binary masking to adjust the relative sparsity of the task-of-interest (denoted with *epsilon*): the performance of the model was studied when the task-of-interest was more sparse than the supplementary tasks ($epsilon \leq 1$) and when it was made less sparse, that is more topics were activated in the task-of-interest than in the supplementary tasks ($epsilon > 1$).

The relative sparsities $epsilon = 0.1, 0.3, 0.75, 1$ and 2 were used. The relative sparsity is controlled using the task-specific ϵ parameters. Technically, when $epsilon \leq 1$, the parameter ϵ_{TOI} controlling the binary masking of the task-of-interest was set to $\epsilon_{TOI} = epsilon$ and the parameters controlling the binary mask-

[1] http://www.gatsby.ucl.ac.uk/~ywteh

ing of the supplementary tasks were all set to the same value $\epsilon_{SUP} = 1$. When $epsilon > 1$, ϵ_{TOI} was set to $\epsilon_{TOI} = 1$ and the masking parameter of all supplementary tasks was set to $\epsilon_{SUP} = \frac{1}{epsilon}$.

The relative sparsity can be interpreted as the ratio of the probabilities of activating topics in the task-of-interest and in the supplementary tasks under otherwise similar conditions except for the value of parameter ϵ. For example, when $epsilon = 0.1$,

$$\frac{p(\text{topic activation in task-of-interest}|\text{data A, parameters B})}{p(\text{topic activation in supplementary tasks}|\text{data A, parameters B})} = 0.1 \qquad (28)$$

where parameters B denote all parameters except for the parameters ϵ of different learning tasks.

The values of the other hyperparameters were fixed: hyperparameter η was fixed to $\eta = 0.00005$ and hyperparameters a_1 and a_2 were fixed to $a_1 = a_2 = 1$.

Toy data was generated as described in section 9.3.1. Dictionary length was fixed to 150. The number of documents in the task-of-interest was set to $16, 32$ or 50 and the number of documents in the 9 supplementary tasks was set to be substantially lower than in the task-of-interest: the number of documents in supplementary tasks was 0.15 or 0.3 times the number of documents in the task-of-interest. The number of words in each document was 15.

The sampler was run for 2000 iterations, and samples 1300 to 2000 were used for predictions after thinning the posterior chain by 6. The experiment was repeated 20 times.

Performance is evaluated as the difference between the predictive likelihoods for test data with learnt and true parameters as described in section 9.1 and averaged over the repetitions. Logarithmic differences to the true model (normalized by the number of words) are reported.

Results are presented in figure 5. As expected, the performance of our model increases when the number of training samples in the task-of-interest increases.

Performance is always best with $\alpha = 3$. Therefore we deduced that in future experiments with this type of toy data the value of α will not be evaluated in a grid, instead a value as small as possible will be used. In future experiments with toy data value $\alpha = 2$ will be used. This specific value is chosen to make α as small as possible while avoiding possible numerical problems induced by having an even smaller value.

The performance of the model decreases as the number of samples in supplementary tasks increases. This implies that the model is not able to make use of information in supplementary tasks efficiently.

According to our hypothesis, the performance should increase when $epsilon \in (0, 1)$ but constraining the sparsity of the task-of-interest excessively should not increase performance. However, performance of our model seems to only increase as $epsilon$ decreases.

Having performance increase even with $epsilon = 0.1$ implies that the observed increase in performance is due to the general resource constraint instead of the

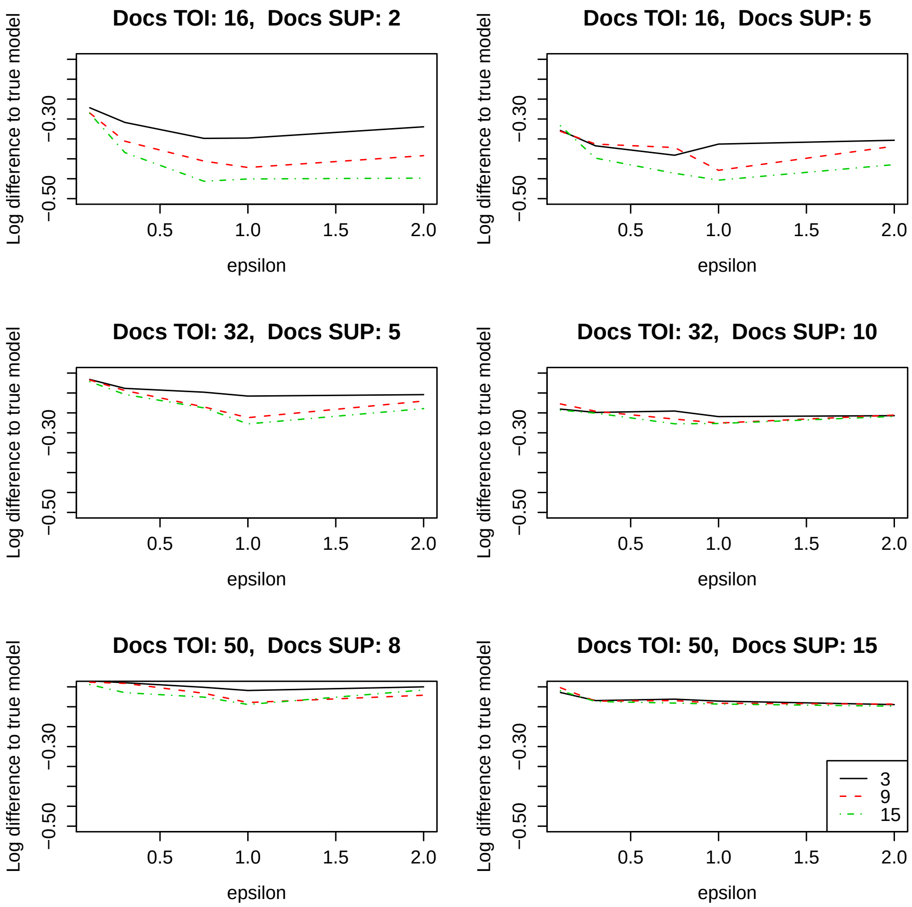

Figure 5: Experiment 1: Performance of our model on toy data with different values of parameter α. The number of documents in the task-of-interest (docs TOI) and 9 supplementary tasks (docs SUP) are varied. The performance of the model is evaluated with different values of α $(3, 9$ and 15 and black solid line, red dashed line and the green dash dot line, correspondingly) and with different relative sparsities *epsilon* $(0.1, 0.3, 0.75, 1, 2)$. Results are presented as the log difference to the true model that is the difference between the average test set log likelihood computed using learnt parameters and true parameters, which has been normalized by the number of words in the test set.

shielding effect. Based on this observation, a general resource constraint is added in the next experiment.

9.5 Experiment 2: Effect of shielding with toy data

In experiment 2, the experimental setup of experiment 1 was reproduced with some changes. The α parameter was not varied in a grid, instead value $\alpha = 2$ was used as the previous experiment demonstrated that large values produce worse results. Now 3000 samples were generated, of which the first 2000 were discarded as burn in and the remaining samples were used for predictions after thinning by 5. Sampling was continued longer because fixing the parameter α reduced the number of MCMC chains needed and more computational resources could be allocated to each of them.

As a conclusion based on the results of experiment 1 we decided to allocate a general resource constraint as described in section 8.2.1. The binary masking was used to increase the penalty for activating topics in all tasks, in other words the ϵ parameters were made smaller than 1 in all tasks.

First ϵ_{TOI} and ϵ_{SUP} were set as described in section 9.4 to produce the relative sparsities $epsilon = 0.1$, 0.3, 0.7, 1, 2 and 4. After this, parameters ϵ_{TOI} and ϵ_{SUP} both were multiplied with 0.1 so that the largest value of the ϵ parameters became 0.1. The value 0.1 was selected by an educated guess.

In addition to using means to describe the results, also quantiles were now used to better describe performance. The experiment was repeated 10 times. Results of experiment 2 are presented in figure 6.

A weak shielding effect can be observed. When the model for the task-of-interest is made more sparse as compared to the models for the supplementary tasks, in other words when $epsilon \in (0, 1)$, the performance in the task-of-interest is slightly better than when the task-of-interest is allocated similar or more resources than the supplementary tasks ($epsilon \geq 1$).According to the hypothesis presented in section 8.1, when resources of the task-of-interest are constrained too much ($epsilon = 0.1$), performance decreases as the model does not allocate sufficient resources to explain the data of the task-of-interest.

Performance in the task-of-interest increases as the number of training documents in the task-of-interest grows. As in experiment 1, as the number of training documents in supplementary tasks increases, performance in the task-of-interest decreases.

9.6 Experiment 3: Assessment of the chosen general resource constraint

The strength of the general resource constraint that was imposed in experiment 2 was chosen based on an educated guess. To further validate this choice, we ran an experiment with a more severe general resource contraint: now the largest value of ϵ was normalized to 0.01.

Other settings in experiment 3 correspond to those in experiment 2. Results of experiment 3 are presented in figure 7.

The shielding effect is not even mildly visible in the results of experiment 3. We interpret this as the overall resource constraint being so strict that it prevents the modeling of the training data properly and causes relatively random results.

9.7 Experiment 4: Effect of shielding with real data and comparison to HDP multi-task and single-task models

Experiment 4 was conducted on real data to study the difference in the performance between our model and the state-of-the-art methods on a real data set.

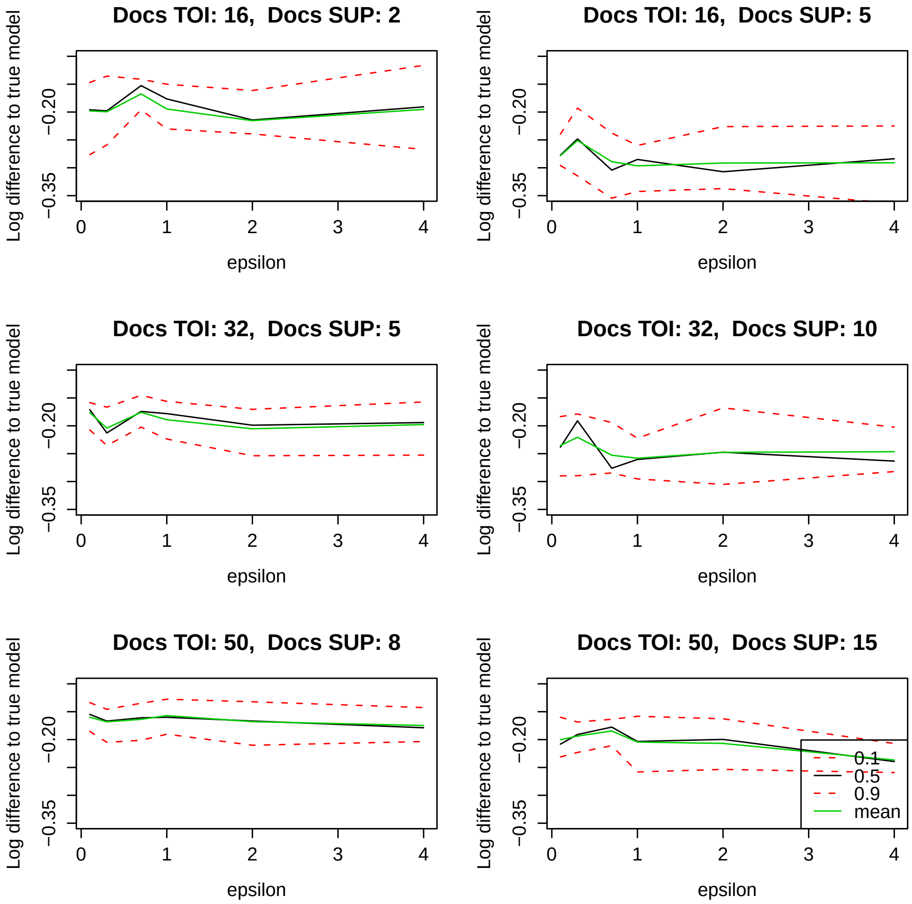

Figure 6: Experiment 2: Performance of our model on toy data with a general resource constraint. Number of documents in the task-of-interest (Docs TOI) and 9 supplementary tasks (Docs SUP) was varied. The value of parameter α was set to 2. The sparsity of the task-of-interest is made different from the other tasks by varying the relative sparsity of the task-of-interest, denoted with *epsilon* $(0.1, 0.3, 0.7, 1, 2, 4)$. In addition to imposing asymmetric sparsity, the binary masking is used to impose a general resource constraint: task-specific ϵ parameters are set so that the largest value of the ϵ parameters is 0.1. Log difference to the true model is the difference between the test data predictive log likelihoods computed using learnt parameters and true parameters, which is normalized by the number of words in the test set. Means (green solid line) and 10%, 50% and 90% quantiles of the log difference to the true model are presented (red dashed line, black solid line and red dashed line, correspondingly).

Our model was compared to a single task method (HDP single task) and a multi-task model (HDP multi-task). For the HDP models, instead of fixing the values of the hyperparameters α_0, α_1 and γ, probability distributions were assigned to these

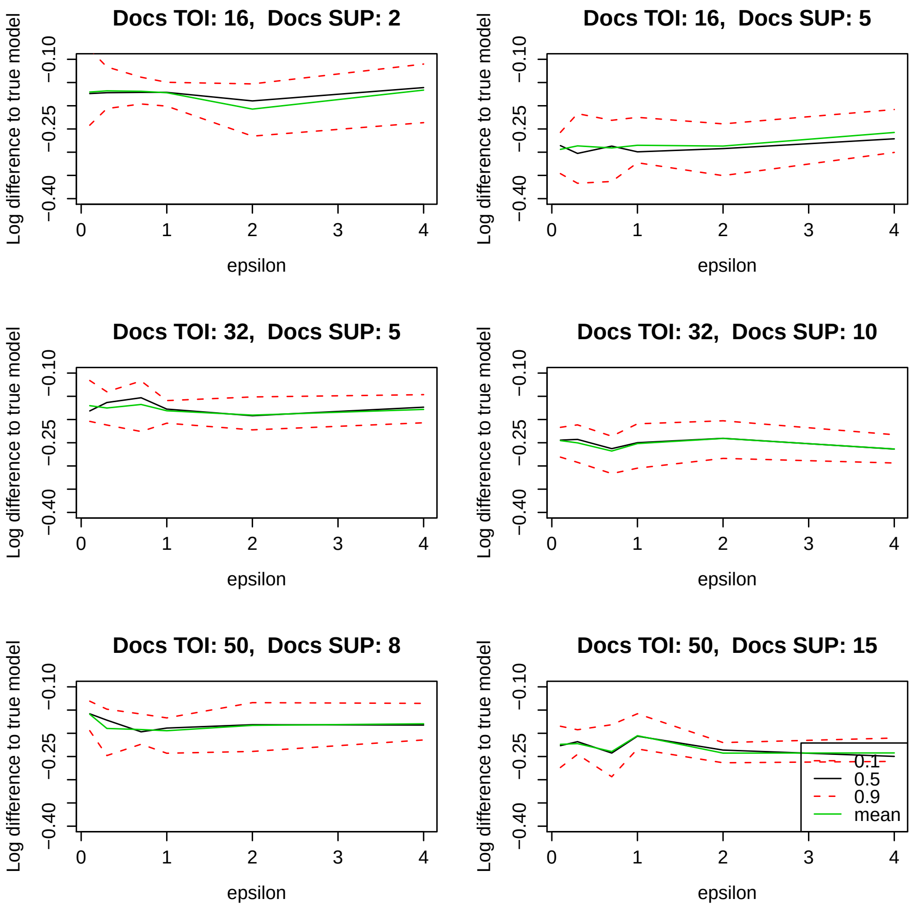

Figure 7: Experiment 3: Performance of our model on toy data with a more strict general resource constraint. Number of documents in task-of-interest (Docs TOI) and 9 supplementary tasks (Docs SUP) was varied. The value of parameter α was set to 2. The sparsity of the task-of-interest is made different from the other tasks by varying the relative sparsity *epsilon* $(0.1, 0.3, 0.7, 1, 2, 4)$. In addition to imposing asymmetric sparsity, the binary masking is used to impose a general resource constraint: task-specific ϵ parameters are set so that the largest value of the ϵ parameters is 0.01. Log difference to the true model is the difference between the test data predictive log likelihoods computed using learnt parameters and true parameters, which is normalized by the number of words in the test set. Means (green solid line) and 10%, 50% and 90% quantiles of the log difference to the true model are presented (red dashed line, black solid line and red dashed line, correspondingly).

hyperparameters and the posterior was integrated over them. The distributions

$$\alpha_0 \sim \text{Gamma}(5, 0.1) \tag{29}$$

$$\alpha_1 \sim \text{Gamma}(0.1, 0.1) \tag{30}$$

$$\gamma \sim \text{Gamma}(5, 0.1) \tag{31}$$

were used as in [11].

The NIPS data set described in section 9.3.2 was used. Different numbers of training documents in the task-of-interest (5, 10, 20 and 40) and in the supplementary task (10 and 25) were used to compare the methods in differing learning setups.

The performance of our model was validated when varying the relative sparsity *epsilon* and the hyperparameter α.

1500 samples were generated from the posteriors of all models. 1000 were discarded as burnin. The parameter η which controls the probability mass assigned a priori to the terms in the dictionary was fixed $\eta = 0.5$ for all models.

Results of experiment 4 are presented in figure 8. Performance of our model is in most cases similar to that of the HDP multi-task model. With very few documents in the task-of-interest (5) and many documents in the supplementary tasks (25), our model performs worse than the HDP multi-task model. Performance of our model is slightly better than that of the HDP single-task model when the number of training documents in the task-of-interest is 10 or less, but as the number of training documents increases, the performance difference vanishes. To study the sigificance of the results, also the standard deviations of the test data perplexity for our model were plotted. The differences between our model and the HDP multi-task model are not significant except for the case in which our model performs poorly (5 documents in the task-of-interest and 25 in the supplementary tasks). The shielding effect is not visible.

The results are good in the sense that having a model perform equally to the state-of-the art method is a success. However, the aim of the experiment was to study the shielding effect, which is not present in the results.

Speculations about possible causes for the observed performance will be discussed further in the following section.

9.7.1 Possible causes for performance differences

The latent structure modeled using the parameters $\mathbf{b}_c$, $\boldsymbol{\phi}_c$ and $\boldsymbol{\psi}_c$ controls the sharing of statistical strength. Limitations of this structure provide a convincing explanation for the absence of the shielding effect in the results with real data.

The latent structure in our model sets penalties for starting new components and sharing old ones. The penalties for activating components control sharing of information and they should enforce sharing of components in a way that improves performance: if new topics can be activated too easily as compared to sharing old ones, all tasks can use a tasks-specific set of components to model their data and no sharing of statistical strength will occur. If sharing is too easy, all tasks will use all components and also such a solution will easily overfit to the training data.

The posterior is affected by both the prior and the likelihood. As the number of samples in training data increases, the relative weight of the prior in the posterior distribution will decrease. This is expectable and rational: the posterior combines the information in the data and in the prior, and as the amount of information contained in the likelihood increases, it should dominate the posterior.

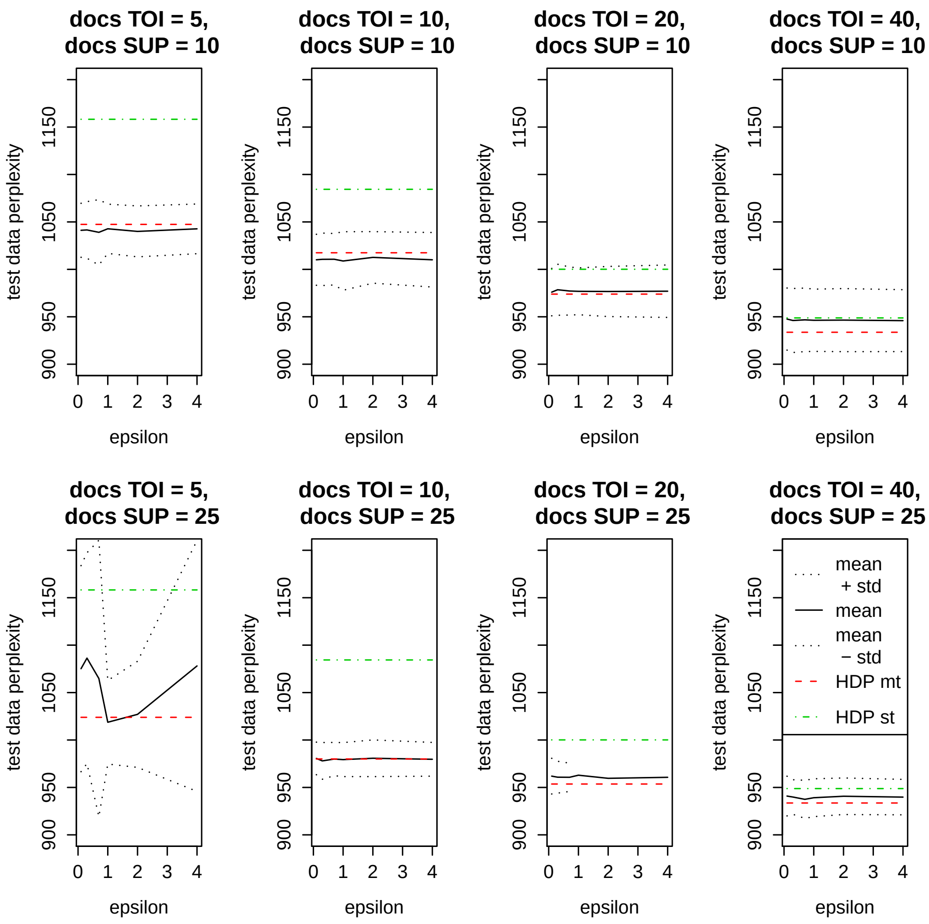

Figure 8: Experiment 4: Comparison of the performance of our model to the HDP single task and multi-task models on the NIPS data set. The figure presents the results using test set perplexities. For our model means (green solid line) and means ± 1 standard deviations are presented (black line and the black dotted lines, correspondingly). For the HDP single-task and multi-task methods, mean test set perplexities are presented (green dashed line and red dash dot line, correspondingly). Number of documents in the task-of-interest (docs TOI) and in the 5 supplementary tasks (docs SUP) is varied. Performance of our model is evaluated with $\alpha = 2, 6, 11$ and different relative sparsities *epsilon* $(0.1, 0.3, 0.7, 1, 2, 4)$. In addition to imposing asymmetric sparsity, the binary masking is used to impose a general resource constraint: task-specific ϵ parameters are set so that the largest value of the ϵ parameters is 0.1.

The challenge with non-parametric models is that the training data likelihood can always be increased by increasing model complexity, in this case by starting new topics and by sharing old ones. Therefore as the amount of training data increases, if the likelihood term dominates the posterior too strongly, the learnt model will become more complex. To prevent overly complex models, the prior for latent structure that promotes sharing should somehow scale up as the amount of data increases.

The most well-established nonparametric models such as the HDP and IBP scale the penalty for adding new components with respect to the amount of data: for example for the IBP, the number of columns C in an IBP matrix (usually used to control model complexity) is proportional to the logarithm of the number of samples, $C \sim \log(N_{observations})$ when the number of rows increases as the amount of data increases [36].

The IBP's automatic scaling of the model complexity to the amount of data (referred to as an *implicit prior for model complexity*) requires that the number of rows grows along with the amount of data. The expected penalties for activating components are inversely proportional to the number of rows in the IBP matrix, $\mathbf{E}(\text{penalty for inverting a 0 into a 1}) \sim \frac{1}{N_{rows}}$, and therefore when the number of rows grows as the amount of data increases, the prior penalties will scale up to compensate for the stronger likelihood term to impose structure.

In [27] the IBP is is used for controlling observation level sharing. The expected penalties for the activation of components are therefore proportional to the number of documents, $\mathbf{E}(\text{penalty}) \sim \frac{1}{N_{documents}}$, whereas the activation allows increasing the likelihood of only one document.

In contrast, we use the IBP to control task-level sharing and the penalties for activating new components are associated to the activation of components in multiple observations simultaenously. As the number of rows in the IBP matrix is now the number of tasks (which is substantially lower than the number of documents), the expected penalties for increasing model complexity, $\mathbf{E}(\text{penalty}) \sim \frac{1}{N_{tasks}}$, are much less significant than when used as in [27]. Simultaneously, the number of observations whose likelihood will be increased by the activation is higher. Therefore it is no wonder that the penalties which should control sharing might become relatively insignificant as the number of documents and words in the tasks increases: the penalties are simultaenously made weaker as the number of observations in the training data whose training likelihoods are increased is made higher.

The NIPS data set contains much more data than the toy data sets with which the shielding effect was observed. The general resource constraint applied in experiment 2 alleviated the problem with toy data. However, the general resource constraint used with the NIPS data was selected on the basis of the results with the toy data and as the NIPS data set contains much more data than the toy data sets, it would probably have required a stronger general resource constraint to constrain overfitting.

Another important aspect of the experimental setup which probably affects the results is the statistical similarity of the learning tasks with the NIPS data. The HDP multi-task model makes a stronger assumption about the similarity of the learning tasks whereas our model is more flexible in this respect. Therefore the HDP probably performs better on data sets in which different learning tasks resemble each other to a larger extent.

With toy data, increasing the number of documents (data) in the supplementary tasks decreased the performance of our model even as approximately 50% of the words in each supplementary task was generated from a topic shared with the task-of-interest. It seems likely that the learning tasks resembled each other: with the NIPS data, the performance of our model increases as the number of documents in supplementary tasks increases. This is evidence for the hypothesis about the similarity of the tasks in the NIPS data set and this probably has favored the HDP multi-task model.

Finally, the sampler used for learning the parameters for the HDP models has been widely used and tested, whereas the sampler for our model has been only used for the purposes of this thesis. It is possible, that the sampler for our model did not cover the posterior as well as the HDP sampler.

The most convincing explanation for the bad performance of the new model seems to be the dominance of the likelihood, which causes severe overfitting. To study this hypothesis, another experiment (experiment 5) was conducted.

9.8 Experiment 5: Assessment of the effect of the size of the data set

The aim of experiment 5 was to study the effect of increasing the amount of data. As the prior of our model controlling information sharing does not scale up as the amount of data (documents, words) in the learning tasks increases, we suspected that as the amount of data increases, the likelihood will dominate and the model will allocate too much resources for modeling the data. When this happens, the structure controlling information sharing will become too weak, and the shielding effect will not emerge.

In experiment 5, the number of words in each document was varied, values $n_{words} = 8, 15, 30, 60$ and 240 were used. The prior controlling information sharing was held constant.

Dictionary length was fixed to 150. The number of documents in the task-of-interest was set to 24 and the number of documents in the 9 supplementary tasks was set to 8, which corresponds to 0.34 times the number of documents in the task-of-interest.

Relative sparsities *epsilon* used were $0.1, 0.3, 0.7, 1$ and 2 and a general resource constraint was enforced by normalizing the ϵ parameters so that the largest value of ϵ became 0.1. Value of α was set to 2 and the value of η was set to 0.00005.

Results of experiment 6 are presented in figure 9. With 8 words per document, the shielding effect is visible. According to our hypothesis, as the number of words in the documents (amount of data) increases, the shielding effect disappears. With higher numbers of words per document, performance increases as *epsilon* is decreased. This suggests that imposing a more strict general resource constraint would be beneficial when the amount of data increases.

An aspect of nonparametric models that we disregarded when formulating the model is the assumption about the infinite data generating process: the HDP and the IBP both assume that an infinite number of components/topics actually exists, and as the number of observations increases, more and more of them will be observed. This means that if we generate data from a parametric distribution with K

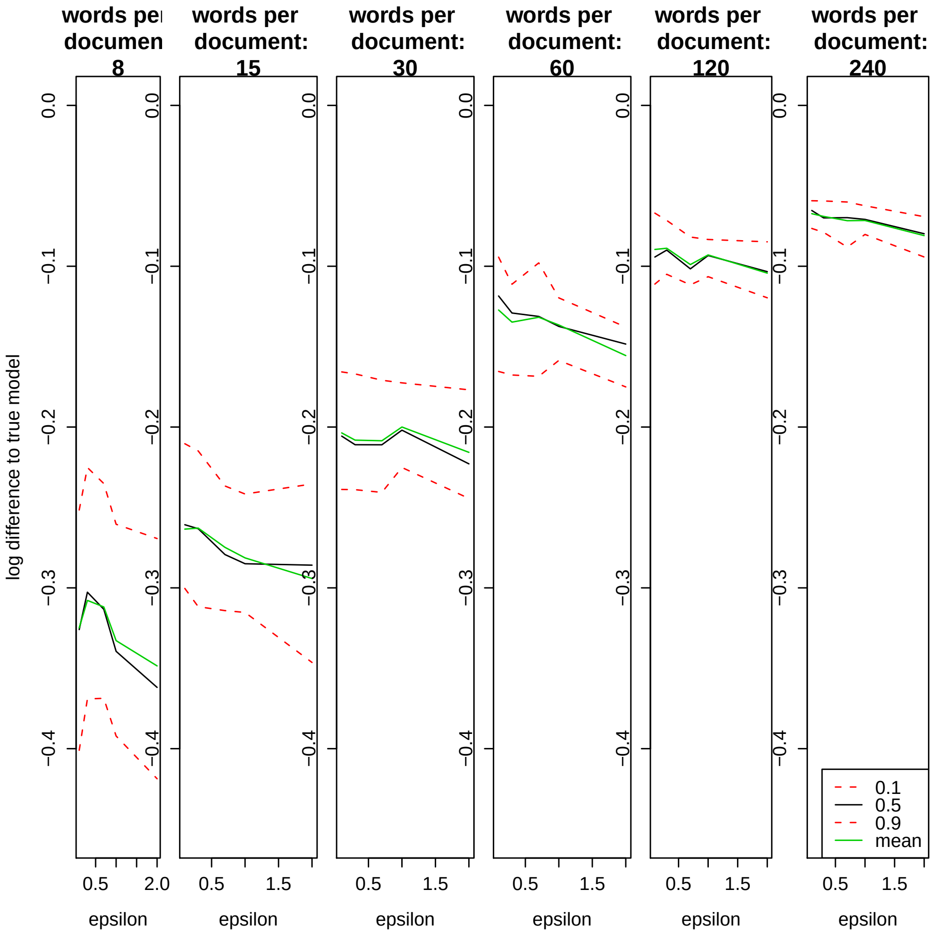

Figure 9: Experiment 5: Effect of the number of words in the training data on the emergence of the shielding effect. Number of words in each document is varied. Performance of our model is evaluated with $\alpha = 2$ and different values of ϵ $(0.1, 0.3, 0.7, 1, 2, 4)$. In addition to imposing asymmetric sparsity, the binary masking is used to impose a general resource constraint: task-specific ϵ parameters are set so that the largest value of the ϵ parameters is 0.1. The number of documents in the task-of-interest is 24 and 8 in the supplementary tasks. Log difference to the true model is the difference between the test data predictive log likelihoods computed using learnt parameters and true parameters, which is normalized by the number of words in the test set. Means (green solid line) and 10%, 50% and 90% quantiles of the log difference to the true model are presented (red dashed line, black solid line and red dashed line, correspondingly).

components and use a nonparametric model to learn parameters, as the amount of data generated from the parametric distribution increases, the nonparametric model will increase the number of components used to model the data regardless of the true underlying parametric distribution. This might be nonintuitive from the traditional Bayesian point of view, in which the posterior of a (parametric) model becomes more peaked around the true data generating parameters when the true data generating distribution belongs to the model family of the model applied. It is difficult to evaluate whether a nonparametric model family is optimal for implementing models that aim to make use of asymmetric sparsity altogether.

To demonstrate this effect, the distribution of the number of components as a function of the number of words in the training documents is presented in figure 10.

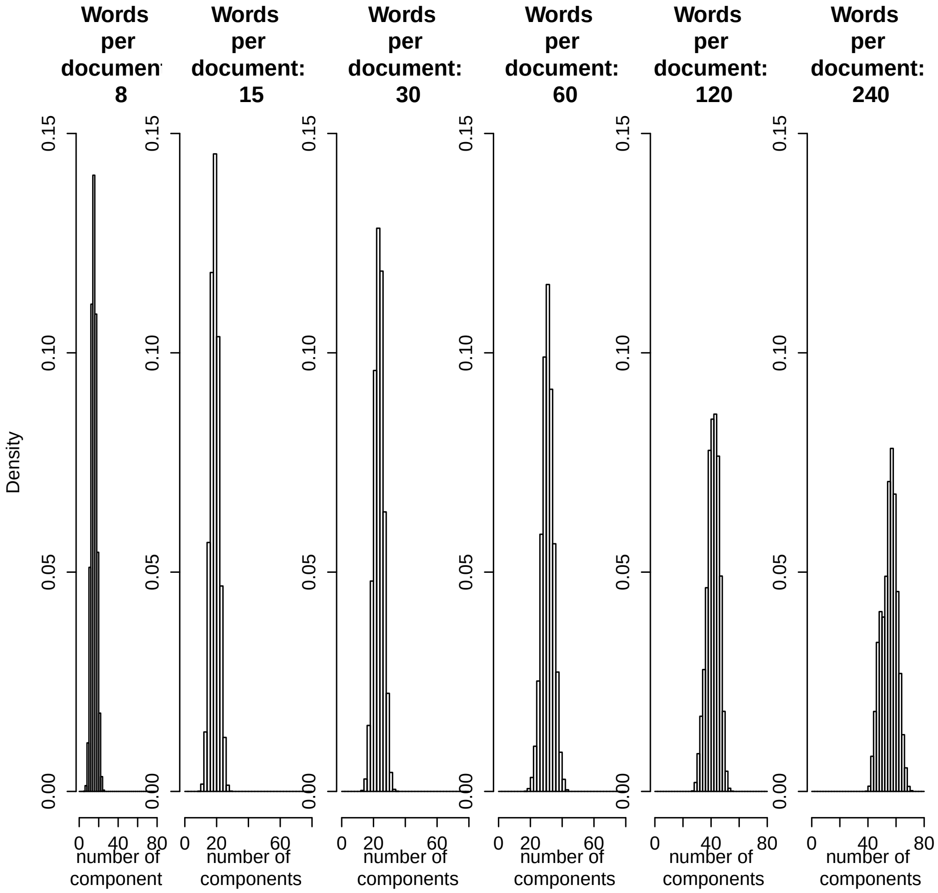

Figure 10: The posterior distribution of the number of components as a function of the number of words in the training documents. The number of documents is held constant. As the number of words in the training documents increases, the posterior of the number of components changes: the expected number of components increases. The data is generated from a parametric distribution with a fixed number of components. The true number of components is 11.

10 Discussion

Multi-task learning, sparsity and topic modeling are active research topics with recent publications in the most recognized machine learning conferences (for example [37], [38], [39] and [40]). In this thesis, multi-task learning using different sparsity constraints for different learning tasks within the topic models framework was studied.

A new Bayesian model referred to as a shielded topic model was presented as a case study of the new approach. The method was designed to utilize the assumed shielding effect to promote the predictive performance in a particularly interesting learning task, which is believed to best resemble future observations, or in other words, the test set.

To be more specific, the new model allows supplementary tasks, which do not perfectly match the distribution of the test set, to use more resources for modeling their data. In other words, the task-of-interest is made more sparse. The excess resources of the supplementary tasks *shield* the shared models for the shared components, and this prevents supplementary-task-specific features from disturbing the learning of the shared characteristics. This should promote performance.

The aim of this thesis was to study the shielding effect and the new model was constructed to study it. The shielding effect was observed with experiments done on toy data. Experiments on real data failed to bring out the shielding effect. Observations about the structure of the new model were made, and they provide hypotheses about the probable causes for the observed behaviour of the model.

Performance of the new model on real data is similar as compared to existing state-of-the-art methods. The new model was compared to single-task and multi-task Hierarchical Dirichlet Process topic models.

The new model does not take into account certain aspects of the Indian Buffet Process prior used in controlling sharing of components and statistical strength. The IBP has been used successfully under circumstances in which also the fixed dimension of the IBP matrix grows as the amount of data increases [27]. This affects greatly important properties of the associated distributions. In the new model, the fixed dimension of the IBP matrix does not necessarily change as the amount of data increases. This prevents our model from scaling up to larger data sets. The inability of the model to scale up as the amount of data increases was studied in the experiments with toy data. In these experiments, the shielding effect disappears as the amount of data increases which is in line with the experiments done on the real data.

A mixture of Gibbs sampling and the Metropolis-Hastings algorithm was used to infer the parameters of the new model. The sampling approach seems to be able to provide estimates for the parameters that produce results that are comparable to comparison methods on a small data set. Therefore, for the comparison with other methods, the sampling approach seems sufficient. However, for experiments using toy data to study the shielding effect, the resolution of the sampler seems insufficient. It is seems probable that the impact of the shielding effect on predictive performance is not drastic. Such delicate effects will easily become unidentifiable when results contain other sources of error.

A question left unanswered by this thesis is, whether the increase in the performance with toy data now ascribed to the shielding effect and asymmetric sparsity could be achieved by a symmetric model by optimizing the general resource constraint relentlessly. This was not studied, as it was considered unlikely.

To further study asymmetric sparsity by using this model, the problem with the inability of our model to scale up to the amount of data should be solved. The binary masking used in our model to impose asymmetric sparsity can be used to alleviate the problem by imposing a general resource constraint, as done in the experiments, but this solution is very inelegant, and more importantly, requires manual work, which is poorly in line with the original motivation for using nonparametric models. A possible solution would be to model the connection between the total number of words in the data and the strength of the general resource constraint. This might allow reducing the amount of manual work. Such solutions would still, however, be very inelegant.

In retrospective, the choice of a slightly modified IBPCD prior as the basis for our model was not a wise one. The prior was used very differently from the setup where it had been shown to be successful. The approach was taken mainly as the prospect of avoiding the assumption in the HDP about the correlation between topic prevalence within the corpus and the topic distribution within each document seemed worthwile (in HDP topics that are active in many documents also are very active within the documents).

For example, the HDP could have been chosen as the basis for studying asymmetric sparsity, which would also have made comparison of performance easier: if all other assumptions (and technical aspects such as implementations) in the models used in the experiments were the same, the effect of asymmetric sparsity would have been the only cause for different performance. A similar binary structure could have been implemented to inactivate some of the topics in the task-of-interest (or supplementary tasks). This would have been justified as the main aim of this thesis was to study the effects of asymmetric sparsity and shielding, not other aspects of the models.

Asymmetric learning has been studied in other publications, for example in [18] and [19]. At the time of the writing of this thesis, the people involved in the research team were not aware of any other approaches directly studying the aspect of using asymmetric sparsity to achieve asymmetric learning.

On real data, evidence for the advantages of asymmetric sparsity were not observed. The model developed was still able to perform equally to the state-of-the-art multi-task model, which must be considered a success. This was not, however, the main objective of this thesis.

The aim of this thesis was to study asymmetric sparsity and the shielding effect by implementing a multi-task model that is able to exploit the assumed advantages of asymmetric sparsity. Shielding effect was produced with toy data, but the model failed to produce effective multi-task performance: even though supplementary tasks contained lots of data from a distribution shared with the task of interest, increasing the amount of overall data in the supplementary tasks caused a decrease in performance. The prior for the latent structure assumed in the model failed to control sharing in an effective manner. If this line of work is continued, some of the modeling assumptions need to be changed.

References

[1] Brusic, V. and Ranganathan, S. *Critical technologies for bioinformatics.* Briefings in Bioinformatics, 2008. Vol. 9, no. 4, pp. 261-262.

[2] Bishop, C. *Pattern Recognition and Machine Learning.* 1. painos, New York, Springer, 2006.

[3] Caruana, R. *Multitask Learning.* Machine Learning, 1997. Vol. 28, pp. 41-75.

[4] Kaski, S. and Peltonen, J. *Learning from Relevant Tasks Only.* In Proceedings of European Conference for Machine Learning (ECML), 2007. Kok, J. N., Koronacki, J., Mantaras, R. L., Matwin, S., Mladeni, D. and Skowron, A. (eds.), Springer, Berlin Heidelberg, 2007, pp. 608-615.

[5] Miller, R. G. *Simultaneous Statistical Inference.* 2nd edition, New York, Springer Verlag, 1981.

[6] Storey, J. D. and Tibshirani, R. *Statistical significance for genomewide studies.* Proceedings of the National Academy of Sciences of the United States of America (PNAS), 2003. Vol. 100, no. 16, pp. 9440-9445.

[7] Gelman, A., Carlin, J. B., Stern, H. S., Rubin D. B. *Bayesian Data Analysis.* 2. painos, Lontoo, Chapman & Hall, 2004.

[8] Krzysztofowicz, R. *Why should a forecaster and a decision maker use Bayes theorem?.* Water Resources Research, 1983. Vol. 19, no. 2, pp. 327-336.

[9] Antoniak, C. E. *Mixtures of Dirichlet Processes With Applications to Bayesian Nonparametrics Problems.* The Annals of Statistics, 1974. Vol. 2, pp. 1152-1174.

[10] Ferguson, T. S. *Bayesian Density Estimation by Mixtures of Normal Distributions.* In Proceedings of Recent Advances in Statistics, 1983. Rizvi, H. and Rustagi, J. (eds.), Academic Press, New York, 1983, pp. 287-303.

[11] Teh, Y., Jordan, M., Beal, M. and Blei, D. *Hierarchical Dirichlet processes.* Journal of the American Statistical Association, 2007. Vol. 101 pp. 1566-1581.

[12] Griffiths, T. L., and Ghahramani, Z. (2005). *Infinite latent feature models and the Indian buffet process.* Technical report no. GCNU TR 2005-001, Gatsby Institute for Computational Neuroscience, University College London.

[13] Teh, Y. W., Gorur, D., and Ghahramani Z. *Stick-breaking construction for the Indian buffet process.* Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS), 2007. Meila, M. and Shen, X. (eds.). Microtome, Brookline, MA, USA, 2007, pp. 556-563.

[14] Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. *Multi-Task Learning for Classification with Dirichlet Process Priors.* Journal of Machine Learning Research, 2007. Vol. 8, pp. 35-63.

[15] Bakker, B. and Heskes, T. *Task Clustering and Gating for Bayesian Multitask Learning.* Journal of Machine Learning Research, 2003. Vol. 4, pp. 83-99.

[16] Wu, P. and Dietterich, T. G. *Improving SVM Accuracy by Training on Auxiliary Data Sources.* In Proceedings of the 21st International Conference on Machine Learning (ICML), 2004. Greiner, R. and Schuurmans, D. (eds.). Omnipress, Madison, WI, 2004, pp. 871-878.

[17] Liao, X., Xue, Y. and Carin, L. *Logistic Regression with an Auxiliary Data Source.* In Proceedings of the 22nd international conference on Machine learning (ICML), 2005. De Raedt, L., Wrobel, S. (eds.). ACM Press, New York, USA, 2005, pp.505-512.

[18] Bickel, S., Bogojeska, J., Lengauer, T. and Scheffer, T. *Multi-Task Learning for HIV Therapy Screening.* In Proceedings of the 25th international conference on Machine Learning (ICML), 2008. McCallum, A and Roweis, S (eds.), ACM New York, NY, USA. 2008, pp. 56-63.

[19] Peltonen, J., Yaslan, Y. and Kaski, S. *Relevant subtask learning by constrained mixture models.* Journal Intelligent Data Analysis, 2010. Vol. 14, pp. 641-662.

[20] Mitchell, J. and Beauchamp, J. *Bayesian variable selection in linear regression.* Journal of the American Statistical Association, 1988. Vol 83, pp. 1023-1036.

[21] Ishwaran, H. and Sunil Rao, J. *Spike and Slab Variable Selection: Frequentist and Bayesian Strategies.* The Annals of Statistics, 2005. Vol. 33, no. 2, pp. 730-773.

[22] Lustig, M., Donoho, D. and Pauly, J. M. *Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging.* Magnetic Resonance in Medicine, 2007. Vol. 58, pp. 1182-1195.

[23] Friedman, J., Hastie, T. and Tibshirani, R. *Sparse inverse covariance estimation with the graphical lasso.* Biostatistics, 2008. Vol 9, pp. 432-441.

[24] Tibshirani R. *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society, Series B, 1996. Vol. 5, paper 8, pp. 267-288.

[25] Obozinski G., Taskar B. and Jordan M. I. *Joint covariate selection and joint subspace selection for multiple classification problems.* Statistics and Computing, 2009. Vol. 20, no. 2, pp. 231-252.

[26] Pontil, M., Argyriou, A. and Evgeniou, T. *Multi-task feature learning.* In Advances in Neural Information Processing Systems (NIPS), 2006. Schölkopf, B., Platt, J. and Hoffman, T.(eds.), MIT Press, Cambridge, MA, 2007, pp. 41-48.

[27] Williamson S., Wang C., Heller K. and Blei D. M. *The IBP Compound Dirichlet Process and its Application to Focused Topic Modeling.* In Proceedings of the 27th International Conference on Machine Learning (ICML), 2010. Fürnkranz, J. and Joachims, T. (eds.), Omnipress, Haifa, Israel, 2010. pp. 1151-1158

[28] Blei, D., Ng, A., and Jordan, M. *Latent Dirichlet allocation.* Journal of Machine Learning Research, 2003. Vol 3, pp. 993-1022.

[29] Caldas J., Gehlenborg, N., Faisal ,A., Brazma A. and Kaski, S. *Probabilistic retrieval and visualization of biologically relevant microarray experiments.* BMC Bioinformatics, 2009. Vol. 25, iss. 12.

[30] Gerber, G. K., Dowell, R. D., Jaakkola, T. S. and Gifford, D. K. *Hierarchical Dirichlet Process-Based Models For Discovery of Cross-species Mammalian Gene Expression Programs.* MIT-CSAIL Technical Report, 2007.

[31] Perina, A., Lovato, P., Murino, V. and Bicego, M. *Biologically-aware latent dirichlet allocation (BaLDA) for the classification of expression microarray.* In Proceedings of the 5th IAPR international conference on Pattern recognition in bioinformatics (PRIB'10), 2010. Tjeerd M. H. Dijkstra, Evgeni Tsivtsivadze, Elena Marchiori, and Tom Heskes (eds.), Springer-Verlag, Berlin, Heidelberg, pp. 230-241.

[32] Gilks, W. R. and Wild, P. *Adaptive rejection sampling for Gibbs sampling.* Applied Statistics, 1992. Vol. 41, pp. 337-348.

[33] Rodriguez, P. P. and Komarek, A. *Package 'ars'.* Updated 17.4.2009. Referred to 25.3.2011. Available from: http://cran.r-project.org/web/packages/ars/.

[34] Li, W. and McCallum, A. *Pachinko allocation: DAG-structured mixture models of topic correlations.* In Proceedings of the 23rd International Conference on Machine Learning (ICML), 2006. Cohen, W., W. and Moore, A. (eds.), ACM, New York, NY, USA, 2006, pp. 577-584.

[35] Wang, C. and Blei, D. M. *Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process.* In Advances in Neural Information Processing Systems (NIPS), 2009. Bengio, Y., Schuurmans, D., Lafferty, J., Williams C., K., I. and Culotta, A.(eds.), MIT Press, Cambridge, MA, 2009, pp. 1982-1989.

[36] Doshi-Velez, F. *The Indian Buffet Process: Scalable Inference and Extensions.* Master's Thesis, University of Cambridge, Department of Engineering, Cambridge, England, 2009.

[37] Lee, S., Zhu, J. and Xing, E. *Adaptive Multi-Task Lasso: with Application to eQTL Detection.* In Advances in Neural Information Processing Systems (NIPS), 2010. Lafferty, J., Williams, C., K. I., Shawe-Taylor, J.(eds.), MIT Press, Cambridge, MA, USA, 2010, pp. 1306-1314.

[38] Zhou, H. and Cheng, Q. *Sufficient Conditions for Generating Group Level Sparsity in a Robust Minimax Framework.* In Advances in Neural Information Processing Systems (NIPS), 2010. Lafferty, J., Williams, C., K. I., Shawe-Taylor, J.(eds.), MIT Press, Cambridge, MA, USA, 2010. Lafferty, J., Williams, C., K. I., Shawe-Taylor, J.(eds.), MIT Press, Cambridge, MA, USA, 2010, pp. 2577-2585.

[39] Jia, Y., Salzmann, M. and Darrell, T. *Factorized Latent Spaces with Structured Sparsity.* In Advances in Neural Information Processing Systems (NIPS), 2010. Lafferty, J., Williams, C., K. I., Shawe-Taylor, J.(eds.), MIT Press, Cambridge, MA, USA, 2010, pp. 982-990.

[40] Hoffman, M., Blei, D. and Bach, F. *Online Learning for Latent Dirichlet Allocation.* In Advances in Neural Information Processing Systems (NIPS), 2010. Lafferty, J., Williams, C., K. I., Shawe-Taylor, J.(eds.), MIT Press, Cambridge, MA, USA, 2010, pp. 856-864.

A Sampling topic assignments

Sampling topic assignments includes an integration that can be seen as the evaluation of the expectation

$$E[\boldsymbol{\theta}_{c,d}|\mathbf{z}_{\backslash c,d,n},\Delta]$$

The expectation is evaluated for each element k one at a time. Using the notation presented in section 8.4.1

$$\begin{aligned}
&E[\theta_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta]\\
\propto&\int_{\theta_{c,d}^{(k)}}\theta_{c,d}^{(k)}\cdot p(\theta_{c,d}^{(k)}|z_{\backslash c,d,n},\Delta)d\theta_{c,d}^{(k)}\\
\propto&\int_{\theta_{c,d}^{(k)}}\theta_{c,d}^{(k)}\cdot\int_{\boldsymbol{\phi}_c}\int_{\mathbf{b_c}}\int_{\Psi_c}\underbrace{p(\theta_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\boldsymbol{\phi}_c,\mathbf{b_c},\boldsymbol{\Psi}_\mathbf{c})}_{\text{posterior of }\theta_{c,d}^{(k)}}\cdot p(\mathbf{b_c^{all}},\boldsymbol{\phi}_\mathbf{c}^\mathbf{all},\boldsymbol{\Psi}_\mathbf{c}^\mathbf{all}|\boldsymbol{\phi}_\mathbf{c}^\bullet,\pi^\bullet,\gamma,\alpha,\epsilon)\\
&d\boldsymbol{\phi}_c d\boldsymbol{b}_c d\boldsymbol{\Psi}_c d\theta_{c,d}^{(k)}
\end{aligned}$$

The posterior of $\theta_{c,d}^{(k)}$ is

$$p(\theta_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\boldsymbol{\phi}_c,\mathbf{b_c},\boldsymbol{\Psi}_\mathbf{c})=\text{multinomial}(\mathbf{z_{c,d,\backslash n}}|\theta_\mathbf{c,d}^\mathbf{(k)})\cdot\text{Dirichlet}(\theta_\mathbf{c,d}^\mathbf{(k)}|\boldsymbol{\phi}_\mathbf{c},\mathbf{b_c},\boldsymbol{\Psi}_\mathbf{c})$$

which is a Dirichlet distribution. The vector $\mathbf{z_{c,d,\backslash n}}$ denotes the topic assignments in document d in task c except for word n. Thus by evaluating the first integral over $\boldsymbol{\theta}_{c,d}^{(k)}$ we get

$$\begin{aligned}
&E[\theta_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta]\\
\propto&\int_{\boldsymbol{\phi}_c}\sum_{\mathbf{b_c}}\sum_{\Psi_c}(n_{c,d,\backslash n}^{(k)}+\phi_c^{(k)})\cdot b_c^{(k)}\cdot\Psi_{c,d}^{(k)}\cdot p(\boldsymbol{b}_c^{all},\boldsymbol{\phi}_c^{all},\Psi_c^{all}|\boldsymbol{\phi}_c^\bullet,\pi^\bullet,\gamma,\alpha,\epsilon,n_{c,d,\backslash n}^{(k)})\\
&d\boldsymbol{\phi}_c
\end{aligned}$$

where $n_{c,d,\backslash n}^{(k)}$ denotes the number of words assigned in document d in task c to topic k without current word n. The integration is only over the unknown elements. In addition to this, only the elements which are nonzero contribute to the expectation. Therefore we get

$$\begin{aligned}
&E[\theta_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta]\\
\propto&\int_{\boldsymbol{\phi}_c^\circ:\boldsymbol{\phi}_c^{\circ(k)}>0}\sum_{\boldsymbol{b}_c^\circ:b_c^{\circ(k)}=1}\sum_{\boldsymbol{\Psi}_c^\circ:\Psi_c^{\circ(k)}=1}(n_{c,d,\backslash n}^{(k)}+\phi_c^{(k)})\cdot b_c^{(k)}\cdot\Psi_{c,d}^{(k)}\cdot\\
&\underbrace{p(\mathbf{b}_\mathbf{c}^\circ,\boldsymbol{\phi}_\mathbf{c}^\circ,\boldsymbol{\Psi}_\mathbf{c}^\circ|\boldsymbol{\phi}_\mathbf{c}^\bullet,\pi^\bullet,\gamma,\alpha,\epsilon,\mathbf{n}_\mathbf{c,d,\backslash n}^\mathbf{(k)})}_{\text{independent of each other given }\theta_{c,d}}d\boldsymbol{\phi}_c^\circ\\
=&\int_{\boldsymbol{\phi}_c^\circ:\boldsymbol{\phi}_c^{\circ(k)}>0}\sum_{\boldsymbol{b}_c^\circ:b_c^{\circ(k)}=1}\sum_{\Psi_c^\circ:\Psi_c^{\circ(k)}=1}(n_{c,d,\backslash n}^{(k)}+\phi_c^{(k)})\cdot b_c^{(k)}\cdot\Psi_{c,d}^{(k)}\cdot\\
&p(\mathbf{b}_\mathbf{c}^\circ|\pi^\bullet,\alpha,)\cdot\mathbf{p}(\boldsymbol{\phi}_\mathbf{c}^\circ|\gamma)\cdot\mathbf{p}(\boldsymbol{\Psi}_\mathbf{c}^\circ|\epsilon)\ \mathbf{d}\boldsymbol{\phi}_\mathbf{c}^\circ
\end{aligned}\tag{32}$$

which is proportional to

$$
\begin{aligned}
&E[\boldsymbol{\theta}_{c,d}^{(k)} | \mathbf{z}_{\setminus c,d,n}, \Delta] \\
\propto &E\left[\frac{(n^{(k)}_{c,d,\setminus c,d,n} + \phi_c^{(k)}) b_c^{(k)} \psi_c^{(k)}}{n^{(.)}_{c,d,\setminus c,d,n} + \sum\limits_j b_c^{(j)}, \psi_c^{(j)}, \phi_c^{(j)}} \right]
\end{aligned}
\tag{33}
$$

The denominator follows from modeling the number of total words using the negative binomial distribution. Equation (33) is evaluated in the following cases:

1. When $n^{(k)}_{c,d,\setminus n} = 0$ and $n^{(k)}_{c,(.),\setminus n} > 0$, we are evaluating the probability of assigning the word n to one of the topics that are currently active in task c. This implies that $\psi_c^{(k)} = b_c^{(k)} = 1$, and Equation (33) becomes:

$$
\frac{(n^{(k)}_{c,d,\setminus n} + \phi_c^{(k)})}{n^{(.)}_{c,d,\setminus n} + \left[\sum\limits_{j:n^{(j)}_{c,(.),\setminus n}>0} \phi_c^{(j)} \right] + \left[\sum\limits_{j:n^{(j)}_{(.),(.),\setminus n}>0} \pi^{(j)}\gamma^{(j)}\epsilon_c \right] + \epsilon_c \cdot a_1 \cdot a_2 \cdot \sum_{n^{(k)}_{(.),(.),(.)}=0} \pi^{(k)}} . \tag{34}
$$

2. When $n^{(k)}_{c,(.),\setminus n} = 0$ and $n^{(k)}_{(.),(.),\setminus n,d,c} > 0$, topic k has not appeared in the current task but it is active in the corpus, so the expectation in Equation (33) becomes

$$
\frac{(\gamma^{(k)}\pi^{(k)}\epsilon_c)}{n^{(.)}_{c,d,\setminus n} + \left[\sum\limits_{j:n^{(j)}_{c,(.),\setminus n}>0} \phi_c^{(j)} \right] + \left[\sum\limits_{j:n^{(j)}_{(.),(.),\setminus n}>0 \,\&\, j\neq k} \pi^{(j)}\gamma^{(j)}\epsilon_c \right] + \gamma^{(k)} + \epsilon_c \cdot a_1 \cdot a_2 \cdot \sum_{n^{(k)}_{(.),(.),(.)}=0} \pi^{(k)}} . \tag{35}
$$

3. When $n^{(k)}_{(.),(.),\setminus n} = 0$, topic k has not appeared anywhere in corpus. The topic can therefore be assigned to any of the currently inactive topics. Therefore in this case we evaluate the probability of assigning any of the infinite number of components:

$$
\frac{\epsilon_c \cdot a_1 \cdot a_2 \cdot \sum_{n^{(k)}_{(.),(.),(.)}=0} \pi^{(k)}}{n^{(.)}_{c,d,\setminus n} + \left[\sum\limits_{j:n^{(j)}_{c,(.),\setminus n}>0} \phi_c^{(j)} \right] + \left[\sum\limits_{j:n^{(j)}_{(.),(.),\setminus n}>0} \pi^{(j)}\gamma^{(j)}\epsilon_c \right] + \epsilon_c \cdot a_1 \cdot a_2 \cdot \sum_{n^{(k)}_{(.),(.),(.)}=0} \pi^{(k)}} . \tag{36}
$$

The stick lengths of the inactive topics with $n^{(k)}_{(.),(.)} = 0$ are distributed according to Equation (20) and their sum $\sum_{k:n^{(k)}_{(.),(.)}=0} \pi^{(k)}$ is evaluated using ARS (adaptive rejection sampling).

B Approximation of the method of empirical likelihoods

Equation 27 differs from standard way of estimating predictive likelihoods with empirical likelihoods. The traditional method approximates the predictive likelihoods as

$$\text{test data predictive likelihoods} \approx \frac{\sum_{documents} \log \left[\frac{1}{S} \sum_{s=1}^{S} \left(\frac{1}{V} \sum_{v=1}^{N_{emp}} \prod_{j=1}^{n_d} p(w|\theta_v) \right) \right]}{\sum_{documents} N_{words,d}}. \tag{37}$$

where S is the number of posterior samples, $\mathbb{I}(\text{word}_{d,w} = term_i)$ is an indicator function used to denote whether $\text{word}_{d,w}$ in the test set corresponds to $term_i$ of the dictionary and N_{emp} is the number of sample topic distribution parameters. If $\text{word}_{d,w} = term_i$, the indicator function has value 1 and 0 otherwise.

The modification presented in section 9.2 makes the approximation

$$\text{mean}(\log(\boldsymbol{x})) \approx \log(\text{mean}(\boldsymbol{x})) \tag{38}$$

which corresponds to a lower bound of the performance as

$$\text{mean}(\log(\boldsymbol{x})) \leq \log(\text{mean}(\boldsymbol{x})). \tag{39}$$

The performance estimate is strongly affected by the posterior samples producing the worst performance: therefore this approximation penalizes such posteriors in which some posterior samples give a very low probability to some words in the test data.

This approximation also disregards the differences between the test documents: the approximation pools all the words in the test documents into one test document and computes model performance on the single pooled document.